

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-316654

(43)Date of publication of application : 16.11.1999

(51)Int.Cl.

G06F 3/06

G06F 3/06

G11B 20/10

(21)Application number : 11-056601

(71)Applicant : HITACHI LTD
HITACHI COMPUT ENG CORP LTD

(22)Date of filing : 04.03.1999

(72)Inventor : AKISAWA MITSURU
KATO KANJI
SUZUKI HIROYOSHI
MAKI TOSHIYUKI

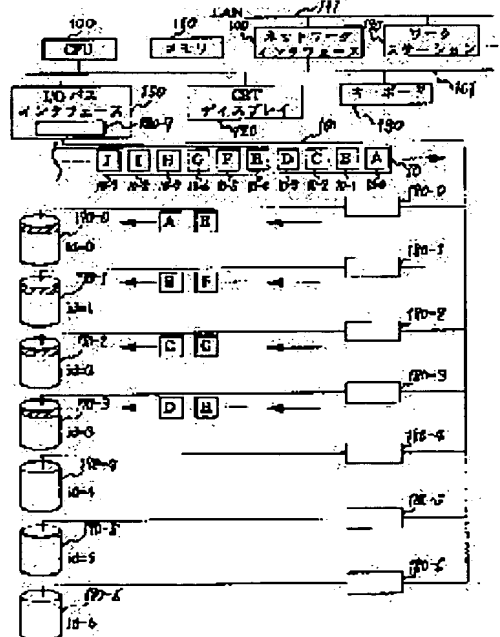
(54) DATA ACCESS CONTROL METHOD, COMPUTER SYSTEM AND DISK ARRAY SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To attain parallel operations of magnetic disk devices and also to attain a fast file access by dividing a file and performing the write and read operations to the different magnetic disk devices.

SOLUTION: In a file read mode, a file 10 stored in a memory 110 is divided and stored in four magnetic disk devices 170-0 to 170-3 (id=0 to 3) with a single data block defined as an I/O unit, for example. Then a data block A10-0 is written into the device 170-0 (id=0).

When the block A10-0 is written into an internal cache of the device 170-0, the next data block B10-1 is written into the device 170-1 (id=1) without waiting for the completion of write of the block A10-0 into a disk medium from an internal cache. Thus, the next write request is issued before a write operation is carried out from an internal cache to a disk medium.



LEGAL STATUS

[Date of request for examination]

04.03.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

3387017

[Date of registration]

10.01.2003

[Number of appeal against examiner's decision]

of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-316654

(43) 公開日 平成11年(1999)11月16日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 F 3/06

3 0 2

G 0 6 F 3/06

3 0 2 D

5 4 0

5 4 0

G 1 1 B 20/10

G 1 1 B 20/10

H

審査請求 有 請求項の数27 O L (全 36 頁)

(21) 出願番号 特願平11-56601
 (62) 分割の表示 特願平4-46685の分割
 (22) 出願日 平成4年(1992)3月4日

(71) 出願人 000005108
 株式会社日立製作所
 東京都千代田区神田駿河台四丁目6番地
 (71) 出願人 000233011
 日立コンピュータエンジニアリング株式会
 社
 神奈川県秦野市堀山下1番地
 (72) 発明者 秋沢 充
 神奈川県川崎市幸区鹿島田890番地株式会
 社日立製作所システム開発本部内
 (74) 代理人 弁理士 小川 勝男

最終頁に続く

(54) 【発明の名称】 データアクセス制御方法および計算機システム並びにディスクアレイシステム

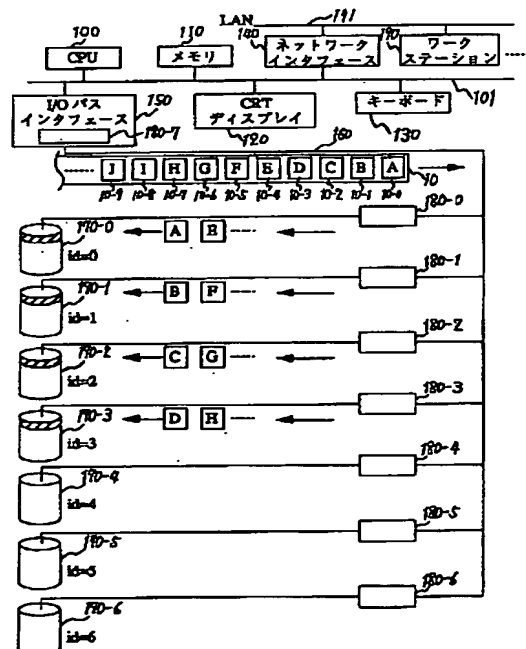
(57) 【要約】

【目的】 本発明の目的は、極めて低コストで高速なファイルアクセスが可能な計算機システムを提供することにある。

【構成】 ディスコネクト・リコネクト機能付き I/Oバスとのインタフェースを有する計算機システム、および I/Oバスに接続された複数の磁気ディスク装置から構成される。計算機システムは、ディスク管理情報、ファイル管理情報、およびファイル記述子対応情報を参照して、ディスクアクセスの際にファイルを分割して複数のディスクに非同期で読み書きするための制御手段を有する。

【効果】 複数の磁気ディスク装置のみで他の特別な制御装置を用いずに、高速なファイルアクセスを実現することができるという効果がある。特殊なハードウェアを必要としないので、従来よりも非常に低コストで高速なファイルアクセスが可能な計算機システムを実現できる。

図1



【特許請求の範囲】

【請求項1】データを複数に分割した第1の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記データを複数に分割した第2の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記第1の分割データが壊れた場合は、上記壊れた第1の分割データに対応する上記第2の分割データを用いて上記第1の分割データを修復することを特徴とするデータアクセス制御方法。

【請求項2】データを複数に分割した第1の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第2の分割データ群を複数の記憶装置のそれぞれに格納し、上記第1の分割データをアクセスする際に、上記第1の分割データが壊れたことを検出した場合は、上記壊れた第1の分割データに対応する上記第2の分割データをアクセスすることを特徴とするデータアクセス制御方法。

【請求項3】データを複数に分割した第1の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第2の分割データ群を複数の記憶装置にそれぞれに格納し、新たなデータを追加する際には、上記新たなデータを複数に分割して上記第1の分割データ群を格納した複数の記憶装置のそれぞれに格納し、上記新たなデータを複数に分割して上記第2の分割データ群を格納した複数の記憶装置のそれぞれに格納することを特徴とするデータアクセス制御方法。

【請求項4】データを複数に分割した第1の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第2の分割データ群を複数の記憶装置にそれぞれに格納し、上記分割されたデータを削除する際には、上記第1の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除し、上記第2の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除することを特徴とするデータアクセス制御方法。

【請求項5】データを複数に分割した第1の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第2の分割データ群を複数の記憶装置にそれぞれに格納し、上記分割されたデータを更新する際には、上記第1の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新し、上記第2の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新することを特徴とするデータアクセス制御方法。

【請求項6】データを第1の記憶装置に格納し、上記データを第2の記憶装置に格納し、第1のデータにアクセスする際に、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データをアクセスするデータアクセス制御方法において、

上記第1の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割し、上記複数の記憶装置のそれぞれに、分割されたデータを格納することを特徴とするデータアクセス制御方法。

【請求項7】データを第1の記憶装置に格納し、上記データを第2の記憶装置に格納し、第1のデータにアクセスする際に、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データをアクセスするデータアクセス制御方法において、

上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割し、上記複数の記憶装置のそれぞれに、分割されたデータを格納することを特徴とするデータアクセス制御方法。

【請求項8】請求項6項のデータアクセス制御方法において、上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割し、上記複数の記憶装置のそれぞれに、分割されたデータを格納することを特徴とするデータアクセス制御方法。

【請求項9】請求項1項から請求項8項のデータアクセス制御方法において、上記記憶装置としてディスク装置を用いることを特徴とするデータアクセス制御方法。

【請求項10】請求項8項のデータアクセス制御方法において、上記第1の記憶装置における複数の記憶装置は、上記第2の記憶装置における複数の記憶装置と同じ記憶装置を用いることを特徴とするデータアクセス制御方法。

【請求項11】データを複数に分割した第1の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数に分割した第2の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記第1の分割データが破壊された場合は、上記破壊された第1の分割データに対応する上記第2の分割データを用いて上記第1の分割データを修復する制御手段とを備えたことを特徴とする計算機システム。

【請求項12】データを格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記データをアクセスする際に、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データをアクセスする制御手段を備えた計算機システムにおいて、

上記第1の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項13】データを格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記データをアクセスする際に、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格

10

20

30

40

50

納している上記データをアクセスする制御手段を備えた計算機システムにおいて、

上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項14】データをに格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データを用いて上記第1の記憶装置に格納した上記データを修復する制御手段を備えた計算機システムにおいて、上記第1の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項15】データを格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データを用いて上記第1の記憶装置に格納した上記データを修復する制御手段を備えた計算機システムにおいて、上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項16】請求項14項の計算機システムにおいて、上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段と、上記第1の記憶装置に格納された上記壊れたデータの壊れた個所に対応する上記第2の記憶装置に格納された分割データを用いて修復する修復手段を備えたことを特徴とする計算機システム。

【請求項17】請求項11項から請求項16項のディスクアレイシステムにおいて、上記記憶装置としてディスク装置を用いることを特徴とする計算機システム。

【請求項18】データを複数の分割した第1の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数の分割した第2の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記第1の分割データが壊れた場合は、上記破壊れた第1の分割データに対応する上記第2の分割データを用いて上記第1の分割データを修復する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項19】データを複数の分割した第1の分割データ群をそれぞれに分配して格納する複数の記憶装置と、

上記データを複数の分割した第2の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記第1の分割データが壊れた場合は、上記壊れた第1の分割データに対応する上記第2の分割データをアクセスする制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項20】データを複数の分割した第1の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数の分割した第2の分割データ群をそれぞれに分配して格納する複数の記憶装置と、新たなデータを追加する際には、上記新たなデータを複数の分割して上記第1の分割データ群を格納した複数の記憶装置のそれぞれに格納し、上記新たなデータを複数の分割して上記第2の分割データ群を格納した複数の記憶装置のそれぞれに格納する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項21】データを複数の分割した第1の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数の分割した第2の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記分割されたデータを削除する際には、上記第1の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除し、上記第2の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項22】データを複数の分割した第1の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数の分割した第2の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記分割されたデータを更新する際には、上記第1の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新し、上記第2の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項23】データを格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記データをアクセスする際に、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データをアクセスする制御手段を備えたディスクアレイシステムにおいて、

上記第1の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項24】データを格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記データをアクセスする際に、上記第1の記憶装置に格納した上記デ

10

20

30

40

50

5

ータが壊れたことを検出した場合、第2の記憶装置に格納している上記データをアクセスする制御手段を備えたディスクアレイシステムにおいて、

上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

- ・【請求項25】データをに格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データを用いて上記第1の記憶装置に格納した上記データを修復する制御手段を備えたディスクアレイシステムにおいて、上記第1の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項26】データを格納する第1の記憶装置と、上記データを格納する第2の記憶装置と、上記第1の記憶装置に格納した上記データが壊れたことを検出した場合、第2の記憶装置に格納している上記データを用いて上記第1の記憶装置に格納した上記データを修復する制御手段を備えたディスクアレイシステムにおいて、上記第2の記憶装置として複数の記憶装置が存在する場合、上記データを複数の分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項27】請求項18項から請求項26項のディスクアレイシステムにおいて、上記記憶装置として磁気ディスク装置を用いることを特徴とするディスクアレイシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、PC、ワークステーションやサーバ等の計算機システム、ディスクアレイシステムに関わり、特に記憶装置に格納されたデータのデータアクセス制御方法および計算機システム並びにディスクアレイシステムに関する。

【0002】

【従来の技術】近年のCPU性能の飛躍的な向上により、ワークステーションやサーバの演算性能は著しく向上した。しかしCPU性能の向上に比較して、I/O性能、特に磁気ディスク装置に格納されたファイルのアクセス速度の向上は十分とは言い難い。これは、磁気ディスク装置のような2次記憶装置のアクセス速度ネック、及びI/Oバスインタフェースの速度ネックが主な要因である。

6

【0003】このネックを解消して高速なファイルアクセスを実現する技術の一つとして、ディスクアレイ装置（もしくはアレイディスク装置と呼ぶ）がある。これは、磁気ディスク装置（以下、簡単のためにディスクドライブあるいは単にディスクと呼ぶ）を複数台内蔵する装置で、各ディスクを並列に動作させることにより、高速なファイルアクセスを実現しようとするものである。さらに、I/Oバスとの接続に高速なインタフェースを用いることにより、I/Oバスインタフェースの速度ネックの解消も図ろうとしている。

【0004】図46にアレイディスク装置の構成を、また図47にファイル読み出しの場合のタイムチャートを示す。図46及び図47は、各スタックが4台のディスクで構成されている例であるが、ディスク台数には増減があっても構わない。これらの図を用いてアレイディスク装置の動作を説明する。

【0005】アレイディスク装置300はホストCPU装置400から読み出し命令を受けると、装置内部のコントローラ（図には示していない）が、まずスタック0のディスクhd0～3にほぼ同時にデータブロック読み出しの起動をかける。スタック0の各ディスクはシーク、回転待ちのあとにディスク媒体から当該データブロックを読み出す。ディスクから読み出されたデータブロック0～3は、それぞれ別々のFIFO（310～313）に格納される。この読み出し動作はスタック0のディスクhd0～3でそれぞれ独立かつ並列に実行されるために、全体の読み出し処理の高速化が実現される。次に、これらのデータブロック0～3は時分割で装置内部の高速バス、すなわちi-Bus320を介してそれぞれバッファ330に順次シーケンシャルに転送される。このバッファ330で各ディスクから独立に読み出したデータブロック0～3を正しい順序に整列する。そして、高速なSCSIバス160を介してデータブロック0～3をホストCPU装置400へ転送する。その後、次の読み出し命令が発行されていれば、データブロックを読み出すべきスタックのディスクに対して、アレイディスク装置内部のコントローラは直ちに読み出しの起動をかける。また、信頼性を確保するために上記データブロックにパリティを用いる方法が知られている。

【0006】アレイディスク装置では、こうした一連の動作を繰り返すことによって、高速なファイルアクセスを実現している。

【0007】なお、アレイディスク装置に関連する公知例としては、特開昭64-41044号「データ転送方式」、特開昭64-21525号「磁気ディスク制御装置」等がある。

【0008】

【発明が解決しようとする課題】上述のように、従来のアレイディスク装置では、パリティを用いて信頼性を確保している。このため上記の4台のアレイディスク例で

は、例えば、ディスクが1台故障した場合、他の3台に格納されている全てのデータをアクセスし、壊れたディスクに格納していたデータの復元を行うため、時間が掛かるという問題がある。

【0009】そこで本発明の目的は、ディスクやデータが壊れた場合、高速に修復するデータアクセス制御方法や計算機システム、ディスクアレイシステムを提供することにある。

・【0010】

【課題を解決するための手段】本発明は上記課題を解決することを目的とし、データを複数に分割した第1の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記データを複数に分割した第2の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記第1の分割データが壊れた場合は、上記壊れた第1の分割データに対応する上記第2の分割データを用いて上記第1の分割データを修復することにより上記課題を解決するものである。

【0011】図48に本発明の原理図を、また図49に本発明におけるファイル読み出しのタイムチャートを示す。図48及び図49はディスク台数が4台の場合を示す例であるが、ディスク台数には増減があっても構わない。これらの図を用いて本発明の原理について説明する。

【0012】本発明においては、計算機システムにSCSIバスのようなディスコネク・リコネク機能を有するI/Oバス160を設けてホストCPU装置400と接続し、さらに、このI/Oバス160に内部キャッシュメモリ20-0～20-3を有する複数台の磁気ディスク装置170-0～170-3を接続し以下のアクセス制御を行う。

【0013】ファイルの書き込みを行う場合には、ホストCPU装置400は該当ファイルをサブファイルに分割して、これらを各磁気ディスク装置170-0～170-3の内部キャッシュメモリ20-0～20-3に次々に書き込んでゆく。ホストCPU装置400がサブファイルを構成するデータブロックを各磁気ディスク装置の内部キャッシュメモリに書き込み終わると、各磁気ディスク装置は書き込み終了をホストCPU装置400へ知らせる。その後、各磁気ディスク装置内部で内部キャッシュメモリに書き込まれたデータが、それぞれシークと回転待ちの後にディスク媒体に書き込まれる。当然この間、ホストCPU装置400は他の磁気ディスク装置の内部キャッシュメモリへ他のサブファイルの書き込みを行うことができる。

【0014】ファイルの読み出しを行う場合には、ホストCPU装置400は、そのファイルを格納するサブファイルを読み出す命令を該当する各磁気ディスク装置170-0～170-3に対して次々に発行してゆく。発行後、I/Oバス160を介したホストCPU装置400

0と各磁気ディスク装置170-0～170-3との接続をディスコネクする。一方、各磁気ディスク装置は読み出し命令を受け付けると、それぞれシークと回転待ちの後にディスク媒体からデータを内部キャッシュメモリへ読み出して、ホストCPU装置400に対してリコネク要求を発行する。最も早くリコネク要求を受け付けられた磁気ディスク装置から、ホストCPU装置400はデータの主記憶装置への読み込みを開始する。この間、他の磁気ディスク装置もホストCPUが発行した読み出し命令に基づき、ディスク媒体から内部キャッシュメモリへデータを読み出す動作を行い、読み出し完了と共にリコネク要求をホストCPU装置400へ発行する。このようにして読み出したサブファイルはホストCPU装置400側で合成され、元ファイルが再生される。

【0015】上述したファイルアクセス動作において、磁気ディスク装置170-0～170-3の内部キャッシュメモリ20-0～20-3へのI/Oバス160を介してのアクセスは高速なI/Oバスの速度で行えるため、シークや回転待ち、およびディスク媒体と内部キャッシュメモリ間のデータ転送時間に比較し、極めて短時間に行えることになる。その結果、読み出し動作も書き込み動作も、ほぼ各磁気ディスク装置がそれぞれ独立に並行して行うことになるため、システム全体としては高速なファイルアクセスが実現されることになる。

【0016】以上に述べたように、ファイルの読み出しと書き込みのどちらの場合でも、磁気ディスク装置の内部キャッシュメモリの存在と、これに対するI/Oバスからのアクセスが高速なことを利用して、磁気ディスク装置の並列動作とI/Oバスの効率的な利用が実現できる。これにより、高速なファイルアクセスを可能とする計算機システムを提供することができる。

【0017】上記の各機能を実現するために、本発明の提供する高速ファイルアクセス制御方法は、ディスコネクおよびリコネク機能を有する少なくとも1本以上の入出力バスを有し、それぞれバスインタフェース装置を介して主記憶装置に接続された該入出力バスに各々異なった装置として識別できる、内部キャッシュメモリを備えた複数の外部記憶装置が接続された計算機システムにおいて、(1)入出力バスに接続された複数の外部記憶装置の中から任意の複数の外部記憶装置を選択し、外部記憶装置グループを定義した外部記憶装置の構成情報を記憶したディスク管理情報を参照して、指定された外部記憶装置グループからこれを構成する外部記憶装置名称を求めるディスク構成情報参照ステップと、(2)アクセスの対象となる元ファイルのファイル記述子とこれを分割構成するサブファイルのファイル記述子とを対応付けて記憶したファイル記述子管理情報を参照して、指定された元ファイルのファイル記述子からこれを構成するサブファイルのファイル記述子を求めるサブファイル

10

20

30

40

50

記述子参照ステップと、(3)元ファイルを構成するサブファイルの、外部記憶装置グループ内の各外部記憶装置上での格納位置情報を記憶したファイル管理情報を参照して、指定されたサブファイル記述子から該サブファイルが格納されるべき、あるいはすでに格納されている該外部記憶装置上の位置情報を求めるファイル位置情報参照ステップと、(4)上記ファイル位置情報参照ステップにより求められた各サブファイルを格納する外部記憶装置名称および該外部記憶装置上の格納位置情報をもとに、ファイル書き込みの際には、書き込み対象となる磁気ディスク装置と当該バスインタフェース装置をコネクして主記憶装置から該当するサブファイルを当該磁気ディスク装置の内部キャッシュメモリに転送し、転送し終えた段階で該内部キャッシュメモリからディスク媒体上への書き込み終了を待たずに、当該磁気ディスク装置をバスインタフェース装置からディスコネクして、次の格納対象となる磁気ディスク装置をコネクして、書き込み対象サブファイルを該磁気ディスク装置の内部キャッシュメモリへ転送し、転送終了次第ディスコネクするという動作を次々に繰り返し、ファイル読み出しの際には、読み出し対象のサブファイルを格納している磁気ディスク装置と当該バスインタフェース装置をコネクして、該磁気ディスク装置に対して読み出し要求を発行し、該磁気ディスク装置内のディスク媒体上のサブファイルが同装置内のキャッシュメモリへ転送されるのを待たずに、該磁気ディスク装置をディスコネクして、直ちに次の磁気ディスク装置とコネクして次のサブファイルの読み出し要求を発行するという動作を繰り返す一方、内部キャッシュメモリ上へサブファイルの読み込みが終了したことを伝えてきた磁気ディスク装置からリコネクを行い、該当キャッシュメモリ上のサブファイルを主記憶上へ読み出すという動作を行うファイルアクセス制御ステップから構成される。

【0018】なお、上記の複数台の磁気ディスク装置から構成される仮想的なディスク装置をバーチャルアレイディスク、または単にバーチャルアレイと呼ぶことにする。

【0019】

【作用】本発明において、上記の課題を解決するための手段で述べた各処理ステップがどのように作用するのかを説明する。

【0020】まず、処理ステップ(1)では、ファイルシステムの指定されたディレクトリにバーチャルアレイディスクを割り当て、利用可能な状態とする。この際に、ディスク管理情報を参照してバーチャルアレイディスクを構成する磁気ディスク装置を認識し、構成要素である磁気ディスク装置をバーチャルアレイディスクとして利用可能な状態にする。

【0021】ディスク管理情報は、バーチャルアレイディスクがどの磁気ディスク装置から構成されるのかを定

義する。一般に、各磁気ディスク装置は複数のパーティションに分割して使用するの、個々の磁気ディスク装置とそのパーティションを指定することで利用する領域を確定できる。ディスク管理情報はバーチャルアレイディスクが使用する磁気ディスク装置とそのパーティションを管理する。

【0022】次に、処理ステップ(2)では、元ファイルのファイル記述子とこれを分割したサブファイルのファイル記述子との対応関係を登録したファイル記述子管理情報を参照して、サブファイルが格納された磁気ディスク装置にアクセスするためのサブファイルのファイル記述子を得る。

【0023】ファイル記述子管理情報は、元ファイルのファイル記述子と、バーチャルアレイディスクに格納したサブファイルのファイル記述子とのあいだの対応を管理する。これにより、サブファイルの格納された複数の磁気ディスク装置にアクセスすることが可能となる。ファイル読み出し、あるいは書き込みの場合には、アプリケーションプログラムから与えられた元ファイルのファイル記述子を、ファイル記述子管理情報を参照してサブファイルのファイル記述子に変換する。また、ファイルを新規に作成して書き込みを行う場合には、新たに元ファイルとサブファイルのファイル記述子を割り当てて、ファイル記述子管理情報にこれらの対応関係を登録する。これ以降、ファイルの分割された実体であるサブファイルへのアクセスは、元ファイルのファイル記述子によって行うことができるようになる。

【0024】次に、処理ステップ(3)では、処理ステップ(2)で得たサブファイルのファイル記述子から、サブファイルのファイル管理情報を得る。これを用いてサブファイルを格納した磁気ディスク装置へアクセスする。

【0025】ファイル管理情報には、サブファイルが格納されている磁気ディスク装置、ストライピングブロック数、サブファイルを構成するデータブロックが格納されている位置を示す位置情報が登録されている。各サブファイルのファイル管理情報を参照して、どの磁気ディスク装置のどこに格納されたデータブロックを順にアクセスするのかを決定する。これにしたがってサブファイルを順にアクセスすることにより、元ファイルをアクセスすることとなる。

【0026】最後に、処理ステップ(4)では、バーチャルアレイディスクに格納されたファイルをアクセスする。実際の処理では元ファイルを格納する磁気ディスクへのアクセスとなる。

【0027】ファイルアクセスは読み出しと書き込みの場合とがあるが、いずれにせよ処理ステップ(3)で得られたサブファイルのファイル管理情報を用いる。ファイル管理情報は、サブファイルの磁気ディスク上での格納位置を管理する。したがって、各サブファイルのファ

10

20

30

40

50

イル管理情報からアクセスするデータブロックの格納位置を得て、次々に各サブファイルを格納する磁気ディスク装置にアクセス要求を発行する。この際に、各サブファイルをアクセスする順番はファイル記述子管理情報から得る。また、サブファイルを格納する各磁気ディスク装置に対して連続してアクセスするブロック数は、ストライピングブロックとしてファイル管理情報に登録されている。

【0028】ファイル読み出しの場合には、読み出し対象のサブファイルが格納されている磁気ディスク装置とI/Oバスインタフェース装置とを接続し、読み出し要求を発行する。要求が発行されると、磁気ディスク装置はヘッドの位置決めを行い、内部キャッシュメモリへディスク媒体からデータを転送する。このために待ち時間が生じる。したがって、磁気ディスク装置とI/Oバスインタフェース装置とを一旦ディスコネクトする。この間に他の磁気ディスク装置とI/Oバスインタフェース装置とを接続し、読み出し要求を発行することができる。この場合にも、やはりヘッドの位置決めと内部キャッシュメモリへのデータ転送を行うための待ち時間が生じるので、さらにまた別の磁気ディスク装置に読み出し要求を発行することができる。ヘッドの位置決めが完了して読み出しが可能となった磁気ディスク装置からデータブロックを読み出し、その直後に次に読み出すデータブロックの読み出し要求を発行し、磁気ディスク装置とI/Oバスインタフェース装置とをディスコネクトするよう制御する。この様にして次々に読み出し要求を各磁気ディスク装置に発行して、サブファイルを主記憶上に読み込むことにより、元ファイルの読み込み処理を行う。

【0029】ファイル書き込みの場合には、格納対象となる磁気ディスク装置とI/Oバスインタフェース装置とを接続し、書き込み要求を発行する。要求発行後にデータブロックを磁気ディスク装置へ転送する。転送されたデータブロックは磁気ディスク装置の内部キャッシュメモリに書き込まれ、ヘッドの位置決めが完了するとディスク媒体へ書き込まれる。この際に、内部キャッシュメモリへのデータブロック書き込みの終了をもって、磁気ディスク装置とI/Oバスインタフェース装置とをディスコネクトし、磁気ディスク装置への書き込みを終了とする。したがって、内部キャッシュメモリからディスク媒体への書き込み終了を待たずに、次の磁気ディスク装置とI/Oバスインタフェース装置とを接続して、書き込み要求発行とデータブロックの転送を行うことができる。この様にして次々に書き込み要求を各磁気ディスク装置へ発行して、元ファイルをサブファイルに分割しつつ主記憶上から書き込むことにより、元ファイルの書き込み処理を行う。

【0030】以上の各処理ステップによるファイルアクセスにおいて、ファイル読み出しの場合には、磁気ディ

スク装置のディスク媒体から内部キャッシュメモリへのデータ転送の間に、次の磁気ディスク装置に読み出し要求を発行できるように、複数の磁気ディスク装置に渡って読み出し処理の多重制御を行うことが可能となる。これにより、各磁気ディスク装置でのシークと回転待ち、および内部キャッシュメモリへのデータ転送が並列に実行される。このため各磁気ディスク装置にサブファイルの読み出し要求を発行し終わると、リコネクト要求をうけて各磁気ディスク装置からデータブロックを読み出す際には、内部キャッシュメモリには主記憶へ読み出し可能なサブファイルのデータブロックが常に存在することとなる。これらを順に読み出しながら次の読み出し要求を発行し、その後に次の読み出し可能な磁気ディスク装置の処理に移って行くように制御することで、I/Oバスの利用効率の高い、高速なファイル読み出しが実現可能となる。

【0031】ファイル書き込みの場合には、磁気ディスク装置の内部キャッシュメモリからディスク媒体へのデータ転送の間に、その終了を待たずに次の磁気ディスク装置に対して書き込み要求を発行できるように、複数の磁気ディスク装置に渡って書き込み処理の多重制御を行うことが可能となる。これにより、各磁気ディスク装置でのシークと回転待ち、および内部キャッシュメモリからディスク媒体へのデータ転送が並列に実行される。このため、各磁気ディスク装置にサブファイルの書き込み要求を順に発行することにより、高速なファイル書き込みが実現可能となる。

【0032】しかも以上のファイルアクセスにおいて、アプリケーションプログラムでは元ファイルのファイル記述子のみを意識すれば良く、サブファイルに分割して格納されていることを意識する必要がない。したがって、単体の磁気ディスク装置に格納されたファイルと同様のアプリケーションインタフェースでアクセスすることが可能である。

【0033】上述のように、本発明では以上の各処理ステップにより複数の磁気ディスク装置を順にアクセスして並列動作させることにより、高速なファイルアクセスを実現する。

【0034】

【実施例】図1に本発明の第1の実施例を示す。本実施例の構成は、共通なデータバス101に接続されたCPU100、メモリ110、CRTディスプレイ120、キーボード130、ネットワークインタフェース140、及びI/Oバスインタフェース150からなる計算機システムである。

【0035】I/Oバスインタフェース150には、I/Oバス160が接続され、このI/Oバス160に7台の磁気ディスク装置170-0～170-6が接続されている。磁気ディスク装置の台数は7台以外の構成も取りうる。各磁気ディスク装置170-0～170-6

とI/Oバス160との間、及びI/Oバスインタフェース150内にはディスコネクト・リコネクト装置180-0~180-7があり、磁気ディスク装置170-0~170-6がI/Oバス160を介してI/Oバスインタフェース150とデータの送受信を行わないときには、両者の電氣的な接続を解放(ディスコネクト)することができる。再び電氣的な接続が必要になったときには、再接続(リコネクト)することができる。これらの制御はCPUと磁気ディスク装置とが協調して行う。なお、上述したディスコネクト・リコネクト装置は、磁気ディスク装置170-0~170-6の内部にディスクコントローラ(図には示していない)とともに内蔵する構成も可能である。

【0036】ネットワークインタフェース140はネットワーク191に接続され、本ネットワーク191に接続された遠隔地のワークステーション190等からのリモートアクセスを可能とする。

【0037】次に本実施例の動作について、まずファイルをディスクに書き込む場合を例に説明する。

【0038】本実施例の計算機システムでは、ファイル10は固定長のデータブロックの集合としてOS(オペレーティングシステム)で管理されている。すなわち、図1に示すように、A10-0、B10-1、C10-2、D10-3、E10-4、F10-5、G10-6、H10-7、I10-8、J10-9、...という一連の複数のブロックから構成される。

【0039】従来技術によるファイル書き込みの場合には、1個の磁気ディスク装置を選択し、その特定の磁気ディスク装置に対して書き込み処理を行う。すなわち、データブロックA10-0、B10-1、C10-2、...を同一磁気ディスク装置、例えば170-0に対して書き込むことになる。その際、データブロックはまず磁気ディスク装置170-0の内部キャッシュメモリ(図示せず)に書き込まれ、シークと回転待ちの後にディスク媒体(図示せず)上に書き込まれる。CPU100は内部キャッシュメモリ(以後、簡単のために内部キャッシュとも表記する)にデータを書き込むと、ディスク媒体への書き込みが終了するまで次の書き込み処理には進まずに待ち状態(wait状態)に入る。したがって、この場合ブロックA10-0のディスク媒体への書き込みが完全に終了して、初めて次のブロックB10-1の書き込みを実行することが可能となる。これは、ブロックB10-1を磁気ディスク装置170-0へ書き込もうとしても、内部キャッシュ上のデータのディスク媒体への書き込みが終了していないため、書き込みが行えないからである。

【0040】一方これに対して、本実施例では複数の磁気ディスク装置にファイルを分割して格納する。例えば、固定長のデータブロックから構成され、メモリ110に格納されているファイル10を、1データブロック

をI/Oの単位として、図1のid=0から3までの4台の磁気ディスク装置170-0~170-3へ分割して格納する。この際、まずデータブロックA10-0をid=0の磁気ディスク装置170-0へ書き込む。データブロックA10-0は磁気ディスク装置170-0の内部キャッシュにまず書き込まれ、次にシークと回転待ちの後にディスク媒体上の所定位置に書き込まれる。これにより書き込み処理が終了することになる。本実施例では、内部キャッシュからディスク媒体への書き込み終了を待つことなく、磁気ディスク装置170-0の内部キャッシュへの書き込みが終了した時点で、次のデータブロックであるブロックB10-1をid=1の磁気ディスク装置170-1へ書き込むという処理を行う。ここでも同様に、データブロックB10-1の内部キャッシュからディスク媒体への書き込み終了を待つことなく、次のデータブロックC10-2をid=2の磁気ディスク装置170-2へ書き込む処理を行う。以下同様にして、データブロックC10-2の内部キャッシュからディスク媒体への書き込み終了を待つことなく、データブロックD10-3をid=3の磁気ディスク装置170-3へ書き込む。データブロックD10-3の磁気ディスク装置170-3の内部キャッシュへの書き込みが終了した後、id=0の磁気ディスク装置170-0へ戻り、内部キャッシュからディスク媒体へのデータブロックA10-0の書き込みが終了していることを確認した上で、ブロックD10-3の書き込み終了を待つことなくブロックE10-4を磁気ディスク装置170-0の内部キャッシュに書き込む。id=0の磁気ディスク装置への書き込みが終了するのに十分な時間が取れない場合には、全体の磁気ディスク装置の台数を増やせば良い。これにより、さらにファイル書き込みの性能を向上させることができる。以下、データブロックF10-5、G10-6、H10-7、I10-8、J10-9、...についても同様の方法で、異なる磁気ディスク装置170-1、170-2、170-3、170-0、...と、順にデータブロックを書き込んでゆく。すべて書き込み終わると、元ファイルの内容を4分割した4個のサブファイルが別々の磁気ディスク装置170-0~170-3に格納されたことになる。

【0041】この様に、磁気ディスク装置の内部キャッシュからディスク媒体へのデータブロックの書き込みの終了を待たずに、次の磁気ディスク装置へのデータブロックの書き込み要求を発行する機能を非同期書き込みと呼ぶ。

【0042】非同期書き込みによりサブファイルを磁気ディスク装置に書き込んでいる時、各磁気ディスク装置170-0~170-6とI/Oバスインタフェース150がI/Oバス160を介して電氣的に接続されるのは、データブロックが磁気ディスク装置の内部キャッシュへ転送される間のみである。すなわち、ホストCPU

10

20

30

40

50

100から書き込み要求が発行されると、磁気ディスク装置がI/Oバスインタフェース150と電氣的に接続され、データブロックの転送が可能となる。磁気ディスク装置の内部キャッシュへのデータブロックの書き込みが終了すると、磁気ディスク装置とI/Oバスインタフェース150との電氣的接続が開放される。その後磁気ディスク装置内部で、内部キャッシュからディスク媒体へのデータ書き込みが独立して行われる。したがって、

【0043】ファイル読み出しの場合についても、本実施例は図1のファイル書き込みの場合と同様に動作する。すなわち、 $id=0$ の磁気ディスク装置170-0から順にデータブロックを読み出していく。ホストCPU100は読み出し命令を磁気ディスク装置170-0から170-1、170-2、...へと順に発行して行く。命令を受け取った磁気ディスク装置はシークと回転待ちを行なうことなく、直ちにディスク媒体からデータブロックを読み込める場合を除き、一旦I/Oバスインタフェース150との電氣的接続を開放する。その後、シークと回転待ちを行なってから、ディスク媒体の所定位置からデータを読み込むシーケンスを開始する。したがって、ホストCPU100は磁気ディスク装置170-0へ読み出し命令を発行した後、直ちに次の読み出し対象となる磁気ディスク装置170-1に対して読み出し命令を発行することが可能となる。

【0044】この様に、磁気ディスク装置からのデータブロックの読み出し終了を待たずに、次の磁気ディスク装置に対してデータブロックの読み出し要求を発行する機能を非同期読み出しと呼ぶ。

【0045】以下、同様にしてホストCPU100は次々に170-2、170-3の磁気ディスク装置へ読み出し命令を発行する。一方、各磁気ディスク装置170-0~170-3の内部では、ディスク媒体から内部キャッシュへデータが読み出されるとホストCPU100に対してリコネクト要求を発行し、I/Oバスインタフェース150との電氣的接続を確立した後ホストCPU100へのデータ転送を行う。これら、ディスク媒体から内部キャッシュへのデータ読み出しは、それぞれの磁気ディスク装置とも相互に独立に、かつ並列に動作することが可能である。すなわち、ファイルの読み出しの場合についても、本実施例では分割したファイルを異なる複数の磁気ディスク装置から読み出すため、磁気ディスク装置の内部キャッシュの効果的な働きにより各磁気ディスク装置の並列動作が可能となる。これにより高速

なファイル読み出しが可能となる。

【0046】以上の説明にあるI/Oバス160を介してのディスコネクト・リコネクト機能は、例えばSCSIインタフェース(Small Computer System Interface、ANSI×3.131-1986規格)にサポートされており、本実施例のI/OバスもSCSIインタフェースの装置を用いることにより実現が可能である。

【0047】図2にファイル読み出し(readと記す場合もある)の場合のタイムチャートを示す。(1)に従来技術である1台の磁気ディスク装置を用いる場合を、(2)に本発明のバーチャルアレイディスクを4台の磁気ディスク装置で構成して用いる本実施例の場合をそれぞれ示す。図中のI/Oバスの軸はI/Oバス160のタイムチャートであり、 id の軸は磁気ディスク装置のタイムチャートである。I/Oバス軸上では、ホストCPU100と磁気ディスク装置170-0~170-3との間で行われる読み出しリクエスト、データ転送のタイミングを示す。各 id 軸上では、ディスク媒体から内部キャッシュへのデータ転送のタイミング、および内部キャッシュからI/Oバス160へのデータ転送のタイミングを示す。

【0048】まず、(1)の1台の磁気ディスク装置の場合について動作を説明する。ホストCPU100から磁気ディスク装置へ読み出し命令が発行され起動が掛かると、磁気ディスク装置内部のコントローラでコマンド解析が行われ、磁気ディスク装置のソフトウェアオーバヘッドT1が生じる。その後、ディスク媒体からデータブロックを読み出すためのヘッドの位置決めが行われ、(シーク時間+回転待ち時間)T2が生じる。この間、磁気ディスク装置はディスコネクトされた状態となる。ヘッドの位置決め終了後、磁気ディスク装置はリコネクトされ、ディスク媒体からデータブロックを内部キャッシュへ読み出しながら、同時に内部キャッシュからI/Oバス160へ出力する。このためデータ転送時間T3が生じる。

【0049】一方、(2)の4台の磁気ディスク装置で構成されるバーチャルアレイディスクの場合には、ホストCPU100から $id=0\sim3$ の4台の磁気ディスク装置へそれぞれ順に読み出し命令が発行され起動が掛かる。

【0050】まず $id=0$ の磁気ディスク装置170-0に読み出し命令が発行されると、コマンド解析後磁気ディスク装置170-0はシークと回転待ちに入り、同時にディスコネクトされる。これによりホストCPU100は次の $id=1$ の磁気ディスク装置170-1に読み出し命令を発行することができるようになる。 $id=1$ の磁気ディスク装置170-1も $id=0$ の磁気ディスク装置と同様に、コマンド解析後シークと回転待ちに入り同時にディスコネクトされる。このとき、 $id=0$ の磁気ディスク装置170-0と $id=1$ の磁気ディスク

17

ク装置170-1はそれぞれ独立に並列動作していることになる。さらにホストCPU100は次のid=2の磁気ディスク装置170-2、id=3の磁気ディスク装置170-3にも同様の手順で読み出し命令を次々に発行する。これらの磁気ディスク装置も同様に、コマンド解析後シークと回転待ちに入り同時にディスコネクトされる。この時点で4台の磁気ディスク装置がそれぞれ全く独立に並列動作することになる。

【0051】id=3の磁気ディスク装置170-3がディスコネクトされた後、id=0の磁気ディスク装置170-0がヘッドの位置決めを完了してデータの読み出しが可能となっていれば、あるいはデータがディスク媒体から読み出されて内部キャッシュに格納されていれば、ホストCPU100に対してリコネクト要求を発行する。ホストCPU100はリコネクト要求を受け付けると、id=0の磁気ディスク装置170-0からデータの読み込みを行う。この際、あらかじめデータが内部キャッシュに格納されていれば、ディスク媒体からの読み出し時間T3より短い時間T4(T3>T4)で高速にデータを読み出すことができる。読み込みが終了すると、id=0の磁気ディスク装置170-0に対して次の読み出し命令を発行する。id=0の磁気ディスク装置170-0は先程と同様に、コマンド解析後シークと回転待ちに入り同時にディスコネクトされる。この時、id=1の磁気ディスク装置170-1が読み出し可能となっていれば、id=0の磁気ディスク装置170-0の場合と同様にホストCPU100は磁気ディスク装置170-1からのリコネクト要求を受け付けて、データを読み込む。読み込みが終了すると、id=1の磁気ディスク装置170-1に対して次の読み出し命令を発行する。そして、本磁気ディスク装置170-1もコマンド解析後シークと回転待ちに入り同時にディスコネクトされる。さらにホストCPU100は次のid=2の磁気ディスク装置170-2、及びid=3の磁気ディスク装置170-3に対しても同様の手順でリコネクト要求を受け付け、データ読み込み、及び次の読み出し命令の発行を行う。以後、このシーケンスが図2に示されるように繰り返される。なお、本図では(シーク時間+回転待ち時間)T2を定数と仮定しているが、実際は各読み出し命令発行の都度異なることが多いような場合には、最も早くリコネクト要求を出してきた磁気ディスク装置から順にデータを読み込むということも可能である。その際に、データを読み出す磁気ディスク装置を選択する順序が前後することになるが、それ以外の上記の手順は変わらない。図2では、id=0の磁気ディスク装置170-0のみがディスク媒体から直接データ転送を行い、他のidの磁気ディスク装置170-1~170-3は内部キャッシュからデータ転送を行うかたちになっている。このためデータ転送時間に差が生じている。これはすなわち、ホストCPU100がデータを読

み込むまでに内部キャッシュへデータが読み込まれている場合にはデータ転送時間は短くなり、読み込まれていない場合にはディスク媒体からの読み出しとなりデータ転送時間は短くはならないためである。

【0052】図2からも明らかなように、本発明では常に4台の磁気ディスク装置が並列に動作しており、単位時間あたりのホストCPU100のデータ読み込み量が実効的に磁気ディスク装置の台数分倍増するという効果が得られる。すなわち、本発明によればファイルの読み出し速度がほぼ磁気ディスク装置の台数分高速化されることになる。

【0053】図2で示したファイル読み出し(read)の、より一般化した制御フロー(1)~(8)を図3に示す。使用する磁気ディスク装置n台はあらかじめ指定されており、これらにファイルが分割して格納されているものとする。これらのファイル読み出しに必要な情報は、例えばアプリケーションプログラムから与えるものとする。また、ファイルアクセスのシーケンス、および非同期読み出しの制御はホストCPU100が行う。

【0054】以下、制御フローの各ステップについて説明する。

【0055】(1)id=0~(n-1)の磁気ディスク装置に対して読み出し要求(readリクエスト)を順次発行する。

【0056】(2)いずれかの磁気ディスク装置が読み出し可能状態となるまで待つ。

【0057】(3)読み出し可能な磁気ディスク装置があれば次のステップ(4)へ進み、なければステップ(2)へ戻る。

【0058】(4)読み出し可能な磁気ディスク装置のid番号をチェックし、記憶する。

【0059】(5)ステップ(4)でチェックしたid=kの磁気ディスク装置から、データを読み出す。

【0060】(6)読み出し終了であればフローを終了し、読み出し終了でなければ次のステップ(7)へ進む。

【0061】(7)id=kの磁気ディスク装置からさらにデータを読み出すのであれば次のステップ(8)へ進み、もうデータを読み出さないのであればステップ(2)へ戻る。

【0062】(8)id=kの磁気ディスク装置に対して読み出し要求(readリクエスト)を発行し、ステップ(2)へ戻る。

【0063】以上の制御フローにより、バーチャルアレイディスクからの高速なファイル読み出しを行うことができる。

【0064】図4にファイル書き込み(writeと記す場合もある)の場合のタイムチャートを示す。(1)に従来技術である1台の磁気ディスク装置を用いる場合を、(2)に本発明のバーチャルアレイディスクを4台の磁気ディスク装置で構成して用いる場合をそれぞれ示す。図中の

I/Oバスの軸、i dの軸の意味はそれぞれ図2と同様である。

【0065】まず、(1)の1台の磁気ディスク装置の場合について動作を説明する。ホストCPU100から磁気ディスク装置へ書き込み命令が発行され起動が掛かると、磁気ディスク装置内部のコントローラでコマンド解析が行われ、磁気ディスク装置のソフトウェアオーバヘッドT1が生じる。この時に、ホストCPU100は書き込むデータを磁気ディスク装置へ転送し、データは直ちに内部キャッシュへ格納される。この時データ転送時間T4が生じる。これをもって、ホストCPU100からのデータ書き込み処理は終了したものとみなされる。その後、磁気ディスク装置ではディスク媒体へデータブロックを書き込むためのヘッドの位置決めが行われ、(シーク時間+回転待ち時間)T2が生じる。この間、磁気ディスク装置はビジー状態となる。ヘッドの位置決め後、磁気ディスク装置はデータブロックを内部キャッシュからディスク媒体へ書き込む。このためデータ転送時間T3が生じる。

【0066】一方、(2)の4台の磁気ディスク装置で構成されるバーチャルレイディスクの場合には、ホストCPU100からi d=0~3の4台の磁気ディスク装置170-0~170-3へ、それぞれ順に書き込み命令が発行されて起動が掛かる。

【0067】まずi d=0の磁気ディスク装置170-0に書き込み命令が発行され、引き続き書き込むデータが転送されて磁気ディスク装置の内部キャッシュへ格納される。これにより、ホストCPU100からのデータ書き込み処理は終了したものとみなされる。磁気ディスク装置はコマンド解析後、シークと回転待ちに入る。

【0068】ヘッドの位置決めが完了すると、内部キャッシュのデータがディスク媒体に書き込まれる。一方、ホストCPU100は次のi d=1の磁気ディスク装置170-1に書き込み命令を発行し、引き続き書き込むデータを転送することができるようになる。i d=1の磁気ディスク装置170-1もi d=0の磁気ディスク装置170-0と同様に、コマンド解析後シークと回転待ちに入り内部キャッシュからディスク媒体へのデータ書き込みを行う。このとき、i d=0の磁気ディスク装置170-0とi d=1の磁気ディスク装置170-1は各々独立に並列動作していることになる。さらにホストCPU100は次のi d=2の磁気ディスク装置170-2、i d=3の磁気ディスク装置170-3にも同様の手順で書き込み命令の発行とこれに引き続くデータ転送を次々に実行する。これらの磁気ディスク装置も同様に、コマンド解析後シークと回転待ちに入り、ヘッドの位置決めが完了すると、内部キャッシュのデータをディスク媒体に書き込む。この時点で4台の磁気ディスク装置170-0~170-3が、それぞれ独立に並列動作することになる。

【0069】i d=3の磁気ディスク装置170-3に書き込み命令の発行と書き込みデータの転送を行った後、i d=0の磁気ディスク装置170-0のデータ書き込みが完了していれば、次のデータの書き込みが可能である。そこで、次の書き込み命令の発行とデータ転送とを実行する。i d=0の磁気ディスク装置170-0は先程と同様にコマンド解析後シークと回転待ちに入る。この時、i d=1の磁気ディスク装置170-1のデータ書き込みが完了していれば、次のデータの書き込みが可能である。そこでi d=0の磁気ディスク装置170-0の場合と同様に、ホストCPU100は磁気ディスク装置へ次の書き込み命令の発行とデータ転送とを実行する。そして、コマンド解析後シークと回転待ちに入る。さらにホストCPU100は次のi d=2の磁気ディスク装置170-2、i d=3の磁気ディスク装置170-3にも前に発行したデータ書き込みの完了を確認すると、同様の手順で次の書き込み命令の発行を行う。以後、このシーケンスが図4に示されるように繰り返される。なお、本図では(シーク時間+回転待ち時間)T2を定数と仮定しているが、実際は各書き込み命令の発行の都度異なることが多い。そのような場合には、最も早くビジー状態が解除されて書き込み可能な状態となった磁気ディスク装置から順にデータを書き込むということも可能である。

【0070】データを書き込む磁気ディスク装置を選択する順序が前後することになるが、それ以外の上記の手順は変わらない。

【0071】図4からも明らかなように、本発明では常に4台の磁気ディスク装置が並列に動作しており、単位時間あたりのホストCPU100のデータ書き込み量が実効的にディスク台数分倍増するという効果が得られる。すなわち、本発明によればファイル書き込み速度が磁気ディスク装置の台数分高速化されることになる。

【0072】図4で示したファイル書き込み(write)の、より一般化した制御フロー(1)~(7)を図5に示す。使用する磁気ディスク装置n台はあらかじめ指定されており、これらにファイルを分割して格納するものとする。これらのファイル書き込みに必要な情報は、例えばアプリケーションプログラムから与えるものとする。また、ファイルの複数磁気ディスク装置への分割、ファイルアクセスのシーケンス、および非同期書き込みの制御はホストCPU100が行う。

【0073】以下、制御フローの各ステップについて説明する。

【0074】(1)i d=0~(n-1)の磁気ディスク装置に対して以下の処理を順次実行する。 1. 書き込み要求(writeリクエスト)を発行する。

【0075】2. 書き込みデータを転送する。

【0076】(2)いずれかの磁気ディスク装置が書き込み可能状態となるまで待つ。

【0077】(3)書き込み可能な磁気ディスク装置があれば次のステップ(4)へ進み、なければステップ(2)へ戻る。

【0078】(4)書き込み可能な磁気ディスク装置の i d 番号をチェックし、記憶する。

【0079】(5) i d = k の磁気ディスク装置にさらにデータを書き込むのであれば、次のステップ(6)へ進み、もうデータを書き込まないのであればステップ(2)へ戻る。

【0080】(6) i d = k の磁気ディスク装置に対して以下の処理を順次実行する。

【0081】1. 書き込み要求(writeリクエスト)を発行する。

【0082】2. 書き込みデータを転送する。

【0083】(7)書き込み終了であればフローを終了し、書き込み終了でなければステップ(2)へ戻る。

【0084】以上の制御フローにより、バーチャルアレイディスクへの高速なファイル書き込みを行うことができる。

【0085】以上に説明したように本実施例ではファイルを分割して異なる複数の磁気ディスク装置への書き込み、および読み出しを行うことにより、各磁気ディスク装置の並列動作が可能となり高速なファイルアクセスが実現される。また、アレイディスク装置のように専用のハードウェアを必要とせず、例えば S C S I バスを用いることにより、7台までの磁気ディスク装置を1枚のアダプタボードで接続することが可能となる。このため、非常に低コストで高速なファイルアクセス機能を実現することができる。

【0086】また、S C S I インタフェースの仕様を拡張すれば、7台以上の磁気ディスク装置を接続することも可能であることは明らかである。さらに、本実施例では固定長のデータブロックを各磁気ディスク装置に分散格納する場合を例にして説明したが、可変長のデータブロックでも同様の処理が可能である。

【0087】図6に本発明の第2の実施例を示す。第2の実施例は、ファイルが複数のサブファイルに分割されて別々の磁気ディスク装置に格納されていることをアプリケーションプログラムから意識しなくとも、ファイルにアクセスできるようにした実施例である。

【0088】以下の各テーブルとプログラムとを用いて、バーチャルアレイディスクとして使用する磁気ディスク装置の指定、ファイルの複数サブファイルへの分割と磁気ディスク装置への対応付け、複数のサブファイルを一まとまりのファイルに見せるアクセス制御、および非同期読み出しと非同期書き込みの制御を行う。

【0089】ディスク管理情報を格納するためのディスク管理テーブル210、ファイル管理情報を格納するためのファイル管理テーブル220、ファイル記述子対応情報を格納するためのファイル記述子対応テーブル23

0をメモリに保持する。アクセス制御プログラム200は、バーチャルアレイディスクを対象とするファイルの書き込みと読み出しの際に、上記各テーブルを参照してアクセス制御を行う。

【0090】上記各テーブル、およびプログラムは計算機システムのメモリに格納される。

【0091】なお、上記の各テーブルはアプリケーションプログラムから与えることも、OSの内部の管理テーブルとして管理することも、いずれの方法も可能である。

【0092】各テーブルの具体的な構成を以下に説明する。

【0093】図7はディスク管理テーブル210の説明図である。ディスク管理テーブルには、どの磁気ディスク装置のどのパーティション(分割領域)を用いてバーチャルアレイディスクを構成するのかということに関する情報が格納される。

【0094】各磁気ディスク装置には、S C S I インタフェースで磁気ディスク装置の識別に用いる i d 番号と、アプリケーションソフトウェアが磁気ディスク装置の識別に用いる磁気ディスク装置名称が割り当てられる。また、通常磁気ディスク装置は複数のパーティションに分割して使用するので、各パーティションにはパーティション番号が付けられる。したがって、i d 番号または磁気ディスク装置名称とパーティション番号との組合せで、システム内の磁気ディスク装置を利用する際の単位領域の指定を行うことができる。これを“論理ディスク装置”と呼ぶことにする。本図に示した例では、ディスク装置名称 h d 6 のディスク装置の各パーティションが論理ディスク装置として定義されており、例えば第5パーティションは名称が h d 6 5 の論理ディスク装置となる。この論理ディスク装置を複数組み合わせ、 “論理バーチャルアレイディスク装置”として使用する。

【0095】ディスク管理テーブル210は磁気ディスク装置の i d 番号または名称とパーティション番号とのマトリクスであり、これらのどの組合せの論理ディスク装置が論理バーチャルアレイディスク装置として定義されているのかを示している。

【0096】名称が h d 0 から h d 3 の4台の磁気ディスク装置をバーチャルアレイディスク装置 v a 0 として定義することとし、その各第0パーティションを図1または図6に示すような論理バーチャルアレイディスク装置 v a 0 0 として用いる場合には、v a 0 0 0 から v a 0 0 3 までを識別子として図7に示すように書き込む。これにより磁気ディスク装置 h d 0 から h d 3 の第0パーティションを論理バーチャルアレイディスク装置 v a 0 0 として定義したことになる。

【0097】ここで識別子 v a X Y Z (ただし、X、Y、Zは一桁の数字)は、この論理ディスク装置がバー

チャルアレイドиск装置 $v a X$ の構成要素であり、かつバーチャルアレイドиск装置 $v a X$ の第 Y パーティションからなる論理バーチャルアレイドиск装置 $v a X Y$ の構成要素であり、さらにその論理バーチャルアレイドиск装置 $v a X Y$ の第 Z 番目の構成要素となる論理ディスク装置であるということを示している。

【0098】なお、ディスク管理テーブル210の設定は、本計算機システムのシステム構成を定義する時点でシステム管理者が行う。これは、ディスク管理テーブル210の設定内容が、各論理ディスク装置の使用状態、すなわち通常のファイル格納に使われているのか、サブファイルの格納に使われているのかということと整合が取れていなければならないからである。

【0099】以上のディスク管理テーブル210の設定により、後述する $mount$ 処理を論理バーチャルアレイドиск装置 $v a 00$ に対して行くと、 $hd0 \sim hd3$ の4台の磁気ディスク装置の第0パーティションを論理バーチャルアレイドиск装置 $v a 00$ として利用することが可能となる。

【0100】図8はファイル管理テーブル220の説明図である。ファイル管理テーブル220は各サブファイルごとに割り付けられ、それぞれファイル属性領域と、ディスクブロックインデックス領域に分けられる。各サブファイルのファイル記述子がファイル管理テーブル220-0~220-nの先頭を指し示すことにより、ファイル管理テーブル220はファイル記述子と関連付けられている。

【0101】ファイル属性領域は、ファイルタイプ、ファイルサイズ、格納デバイス、およびストライピングブロックを格納するエントリから構成される。ファイルタイプには、このテーブルが管理するファイルがバーチャルアレイドиск装置に格納されるサブファイルであるのか、通常の磁気ディスク装置に格納されるファイルであるのかを示す識別子が格納される。ファイルサイズには、サブファイルの容量が格納される。格納デバイスには、サブファイルが格納される磁気ディスク装置の実体である、論理ディスク装置の名称が格納される。ストライピングブロックには、ファイルをこのサブファイルに分割した際に単位としたデータブロックの個数が格納される。すなわち各サブファイルは、このデータブロックの個数ごとにファイルを先頭から分割して作られる。

【0102】ディスクブロックインデックス領域には、データブロックのインデックス情報としてディスク上の論理ブロック番号を格納する。本図では、ファイルを構成する各データブロックが論理ブロック番号100、200、300、...、900に格納されていることになる。

【0103】ファイル管理テーブル220はバーチャルアレイドискに格納するファイルが作られるときに、アクセス制御プログラムによってサブファイルごとに作

られる。以後、このファイルが消去されるまでファイル管理テーブルは存続し、ファイルを構成する各サブファイルをアクセスする際にアクセス制御プログラムがこれを参照する。バーチャルアレイドискに格納されたファイルが消去されるときに各サブファイルが消去されることになり、この時同時にファイル管理テーブルも消去される。

【0104】図9はファイル記述子対応テーブル230の説明図である。バーチャルアレイドискに格納されている元ファイルのファイル記述子 $v f d$ と、分割後のサブファイルのファイル記述子 $f d$ との対応を示している。

【0105】本図の例では、 $v f d = 4$ である元ファイルは、4個のサブファイルからなり、それぞれのファイル記述子は、 $f d 0 = 5$ 、 $f d 1 = 6$ 、 $f d 2 = 7$ 、 $f d 3 = 8$ である。また、 $v f d = 20$ の元ファイルについても同様に、サブファイルのファイル記述子は $f d 0 = 21$ 、 $f d 1 = 22$ 、 $f d 2 = 23$ 、 $f d 3 = 24$ であることを示している。

【0106】バーチャルアレイドискに格納されているファイルをアクセスする際には、まずファイルを $open$ しなければならない。この時に、元ファイルのファイル記述子 $v f d$ がアクセス制御プログラムにより割り当てられる。次にその実体であるサブファイルの名称が元ファイルの名称から生成され、このサブファイル名称を用いてサブファイルが $open$ される。この時に各サブファイルにファイル記述子 $f d 0 \sim f d 3$ が割り当てられる。以後、アプリケーションプログラムからこのファイルをアクセスする場合にはファイル記述子 $v f d$ を用いる。アクセス制御プログラムはファイル記述子対応テーブル230を用い、これをサブファイルのファイル記述子 $f d 0 \sim f d 3$ に変換して各サブファイルのアクセスを行う。ファイル記述子対応テーブルのエントリは、ファイルを $close$ する際に解放されて内容は無効となる。

【0107】以上説明したディスク管理テーブル210、ファイル管理テーブル220、およびファイル記述子対応テーブル230を用いて、バーチャルアレイドискを制御する際の概略の手順について説明する。

【0108】図10に制御の全体のフローを示す。各ステップは、ユーザからのコマンド、あるいはプログラムからの関数コールによって起動される処理を示している。各処理はメモリに格納されたアクセス制御プログラムによって実行される。以下、本図の制御フローに従ってその内容を説明する。

【0109】 $mount$ 処理では、バーチャルアレイドискをOSが管理するディレクトリの指定された場所に割り付け、その指定ディレクトリ名称下のデバイスとしてソフトウェアから利用可能な状態とする。すなわち、論理バーチャルアレイドиск装置を構成する論理

10

20

30

40

50

ディスク装置を、ディスク管理テーブル210を参照してディレクトリの指定された場所に割り付ける処理を行なう。

【0110】open処理では、元ファイル及びサブファイルにファイル記述子を割り当て、ファイル記述子対応テーブル230に対応関係が取れるように登録する。

・【0111】read/write処理では、read処理指定時にはバーチャルアレイドискからのファイル読み出しを行い、write処理指定時にはバーチャルアレイドискへのファイル書き込みを行う。いずれの場合も、元ファイルのファイル記述子からファイル記述子対応テーブル230によりサブファイルのファイル記述子を得る。さらにファイル記述子によりファイル管理テーブル220を得て、サブファイルの格納されている論理ディスク装置にアクセスする。

【0112】close処理では、元ファイルとサブファイルのクローズ処理、すなわち元ファイルに割り当てたファイル記述子と、サブファイルに割り当てたファイル記述子の解放を行う。

【0113】umount処理では、論理バーチャルアレイドиск装置として指定ディレクトリに割り付けられた各論理ディスク装置を指定ディレクトリから除去する。

【0114】このため、その指定ディレクトリ名称下のデバイスとしてソフトウェアから意識できなくなる。

【0115】以下、各処理の詳細な制御フローを図11、図12、図13、図14、図15、図16、および図17を用いて説明する。

【0116】mount処理の制御フロー(1)～

(3)を図11を用いて説明する。なお、/devで始まる装置名称は、mount処理やumount処理で用いられるフルパス名称である。

【0117】(1)マウントする論理バーチャルアレイドиск装置に対応する磁気ディスク装置の名称、パーティション番号、すなわち論理ディスク装置の名称をディスク管理テーブル210から得る。

【0118】(2)論理バーチャルアレイドиск装置をマウントするディレクトリに、マウント用のディレクトリを作る。例えば図12に示すように論理バーチャルアレイドиск装置の名称が/dev/va00である場合には、名称が“.va00"というディレクトリをまずマウントディレクトリ/dataの下に作り、さらにその下に、すなわち/data/.va00の下に名称が“.va000"、“.va001"、“.va002"、および“.va003"というディレクトリを作る。

【0119】(3)ディスク管理テーブル210を参照して、各ディレクトリ“.va000"、“.va001"、“.va002"、および“.va003"に対応する論理ディスク装置/dev/hd00～/dev/

hd03をマウントする。

【0120】図12には、mount処理の結果により構成されるトリ構造が示されている。

【0121】このようにmount処理ではトリ構造を持つディレクトリ下にバーチャルアレイドискがマウントされ、バーチャルアレイドиск内のファイルがトリ構造のディレクトリによって管理される。この例では、“/data”ディレクトリに論理バーチャルアレイドиск装置“/dev/va00”がマウントされている。各磁気ディスク装置には同一構造のディレクトリが構成され、その下に各サブファイルが格納されることになる。ここでは、各サブファイル名称とディレクトリ名称に添字を付して区別してある。

【0122】open処理の制御フロー(1)～(5)を図13を用いて説明する。

【0123】(1)オープン対象として指定された元ファイルの名称から、添字を付してサブファイルの名称を生成する。

【0124】例えば、図12のディレクトリ構造において、元ファイルとして/data/fileを指定すると、これからサブファイルのファイル名称

/data/.va00/.va000/file0、
/data/.va00/.va001/file1、
/data/.va00/.va002/file2、
/data/.va00/.va003/file3、
を生成する。

【0125】(2)サブファイルのファイル名称を用いて個々のサブファイルをオープンし、ファイル記述子fd0～fd3を割当てる。これにより、各サブファイルのファイル記述子fd0～fd3がファイル管理テーブル220に対応付けられ、アプリケーションプログラムからのファイル記述子を用いたアクセスが可能となる。

【0126】(3)元ファイルにファイル記述子vfdを割り当てる。

【0127】(4)元ファイルのファイル記述子vfdとサブファイルのファイル記述子fd0～fd3とを対にして、ファイル記述子対応テーブル230に登録する。(5)元ファイルのファイル記述子vfdを、本open処理の結果として、すなわちopenコールの戻り値としてアプリケーションプログラムへ返す。

【0128】次にread/write処理でのread処理実行の制御フロー(1)～(5)について図14を用いて説明する。

【0129】(1)アプリケーションプログラムからは元ファイルのファイル記述子vfdを引き数にしてアクセス要求が発行される。引数であるファイル記述子vfdをキーとしてファイル記述子対応テーブル230を検索し、vfdに対応するサブファイルのファイル記述子fd0～fd3を得る。図9に示すように例えば元ファイルのファイル記述子vfdが4の場合、サブファイル

10

20

30

40

50

のファイル記述子 $f d 0 \sim f d 3$ として5、6、7、8が得られる。

【0130】(2) 次に、サブファイルのファイル記述子 $f d 0 \sim f d 3$ から、これらのサブファイルのファイル管理テーブル220を得る。

【0131】(3) サブファイルをアクセスする順番は、ディスク管理テーブル210の記述子の添字の順に行う。この順にサブファイルのファイル管理テーブル220を参照し、読み込むデータブロックの論理ブロック番号を得る。図7に示す例では、論理ディスク装置/ $d e v / h d 0 0$ 、/ $d e v / h d 0 1$ 、/ $d e v / h d 0 2$ 、/ $d e v / h d 0 3$ の順にアクセスすることになる。ファイル管理テーブル220に書かれたストライピングブロックが指定するデータブロック数だけアクセスしたら、次のサブファイルへアクセスする。すなわち、次のサブファイルのファイル管理テーブルを参照して、ストライピングブロックが指定するデータブロック数だけアクセスする。この手順にしたがって、読み込むデータブロックの論理ブロック番号を順に得る。

【0132】(4) 磁気ディスク装置から読み込むデータブロックの論理ブロック番号と、サブファイルのファイル管理テーブル220の格納デバイスエントリに書かれた論理ディスク装置とにしたがって、データブロックを読み込む。実際にアクセスする磁気ディスク装置は、ディスク管理テーブル210を参照して、該当論理ディスク装置をその一部として含む磁気ディスク装置として得ることができる。

【0133】(5) 最後に、読み込むべきデータブロックをすべて読み込んだかどうか判定する。まだ読み込むべきデータブロックがあれば、(3)に戻って繰り返す。

【0134】次に $r e a d / w r i t e$ 処理での $w r i t e$ 処理実行の制御フロー(1)～(7)について図15を用いて説明する。サブファイルのデータブロックをアクセスする手順の概略は、 $r e a d$ 処理実行時と同様である。

【0135】(1) アプリケーションプログラムからは元ファイルのファイル記述子 $v f d$ を引き数にしてアクセス要求が発行される。引数であるファイル記述子 $v f d$ をキーとしてファイル記述子対応テーブル230を検索し、 $v f d$ に対応するサブファイルのファイル記述子 $f d 0 \sim f d 3$ を得る。図9に示すように例えば元ファイルのファイル記述子 $v f d$ が4の場合、サブファイルのファイル記述子 $f d 0 \sim f d 3$ として5、6、7、8が得られる。

【0136】(2) 次に、サブファイルのファイル記述子 $f d 0 \sim f d 3$ から、これらのサブファイルのファイル管理テーブル220を得る。

【0137】(3) 既存ファイルの内容を更新する場合には、データブロックの内容を上書きすれば良い。しか

し、ファイルサイズを超えて追加書き込みを行う場合には、書き込むために新しい領域を割り当てる必要が生じる。したがって、新しいディスクブロックの割当てが必要であるかどうかを判定し、必要であれば(4)へ、不要であれば(5)へ進む。

【0138】(4) ディスクブロックを割り当てる際に、どのサブファイルが格納されている論理ディスク装置から割り当てるのかを決める。ファイルの最終部分となっているサブファイルの総データブロック数が、ファイル管理テーブルのストライピングブロックエントリに書かれたデータブロック数の整数倍である場合、次のサブファイルを格納する論理ディスク装置からデータブロックの割り当てを行う。そうでなければ、ファイル最終部分のサブファイルが格納されている論理ディスク装置から割り当てを行う。データブロックを割り当てた後に、そのデータブロックの論理ブロック番号を得る。次は(6)へ進む。

【0139】(5) $r e a d$ 処理の場合で述べたように元ファイルはサブファイルに分割されており、これにより元ファイルに合成するためのアクセスする順番は決定される。したがって、その手順に従い書き込み箇所に該当するサブファイルのファイル管理テーブル220から、ファイルが書き込まれるデータブロックの論理ブロック番号を得る。

【0140】(6) 磁気ディスク装置へ書き込むデータブロックの論理ブロック番号と、サブファイルのファイル管理テーブル220の格納デバイスエントリに書かれた論理ディスク装置とにしたがって、データブロックを書き込む。実際にアクセスする磁気ディスク装置は、ディスク管理テーブル210を参照して、該当論理ディスク装置をその一部として含む磁気ディスク装置として得ることができる。

【0141】(7) 最後に、書き込むべきデータブロックをすべて書き込んだかどうか判定する。まだ書き込むべきデータブロックがあれば、(3)に戻って繰り返す。

【0142】 $c l o s e$ 処理の制御フロー(1)～(3)を図16を用いて説明する。

【0143】(1) アプリケーションプログラムからは元ファイルのファイル記述子 $v f d$ を引き数にして処理要求が発行される。ファイル記述子 $v f d$ をキーとしてファイル記述子対応テーブル230を検索し、 $v f d$ に対応するサブファイルのファイル記述子 $f d 0 \sim f d 3$ を得る。

【0144】(2) サブファイルに対応するファイル記述子 $f d 0 \sim f d 3$ を解放する。

【0145】(3) 元ファイルに対応するファイル記述子 $v f d$ を解放する。

【0146】最後に、 $u m o u n t$ 処理の制御フロー(1)～(3)を図17を用いて説明する。

【0147】(1) ディスク管理テーブル210を参照して、アンマウントする論理バーチャルアレイディスク装置を構成する論理ディスク装置に対応する、磁気ディスク装置の名称とパーティション番号を得る。

【0148】(2) 上記、論理ディスク装置をアンマウントする。すなわち図12の例では、ディレクトリ “.va000”、“.va001”、“.va002”、および “.va003” から /dev/hd00 ~ /dev/hd30 を除去する。

【0149】(3) バーチャルアレイディスクのマウント用に作ったディレクトリを消去する。すなわち図12の例では、 “.va00”、 “.va000”、 “.va001”、 “.va002”、 および “.va003” を消去する。

【0150】以上説明したように、ディスク管理テーブル210によりバーチャルアレイディスクを定義し、ファイル記述子対応テーブル230により元ファイルとサブファイルとの対応付けを行い、ファイル管理テーブル220を参照してサブファイルへアクセスすることにより、ユーザは複数のサブファイルにアクセスすることを全く意識せず、あたかも単一のファイルにアクセスするのと全く同じ形でバーチャルアレイディスクを利用することができるという効果が得られる。

【0151】次に図18~21を用いてファイルのストライピング処理について説明する。

【0152】図1に示した実施例では、データブロック1個を単位としたファイルのサブファイルへの分割、すなわちストライピングを行っている。この場合、データブロックAから順にB、C、Dを、磁気ディスク装置のid番号0番から順に1番、2番、3番へと格納する。そして、次のデータブロックEから順にF、G、Hを、同様に磁気ディスク装置のid番号0番から順に1番、2番、3番へと格納する。この場合のファイルの書き込みの様子を図18に、ファイルの読み出しの様子を図19に示す。

【0153】図18はファイルを先頭から順に書き込む場合を示している。I/Oバスインタフェース150を介して、ホストCPU100からデータブロックが順に送られる。ディスコネクト・リコネクト機能により各磁気ディスク装置がI/Oバスインタフェース150と順に接続されて、データブロック0、1、2、3、4... がid番号0番、1番、2番、3番、0番... の磁気ディスク装置へと格納される。なお、格納する磁気ディスク装置の順序は必ずしもid番号の順である必要はなく、0番以外の磁気ディスク装置から開始することも可能である。また、id番号の順序を昇順、または降順以外とすることも可能である。

【0154】図19はファイルを先頭から順に読み出す場合を示している。I/Oバスインタフェース150を介して、ホストCPU100へデータブロックが順に送

られる。図18に示したようにデータブロックは格納されており、ディスコネクト・リコネクト機能により各磁気ディスク装置がI/Oバスインタフェース150と順に接続されて、データブロック0、1、2、3、4... がid番号0番、1番、2番、3番、0番... の磁気ディスク装置から読み出される。なお、格納する磁気ディスク装置の順序は必ずしもid番号の順である必要はなく、0番以外の磁気ディスク装置から開始することも可能である。また、id番号の順序を昇順、または降順以外とすることも可能である。

【0155】一方、複数のデータブロックをストライピングの単位とすることも可能である。一例として、データブロック4個を単位としてストライピングを行った場合のファイルの書き込みの様子を図20に、ファイルの読み出しの様子を図21に示す。

【0156】図20はデータブロック4個を単位としてファイルを先頭から順に書き込む場合を示している。I/Oバスインタフェース150を介して、ホストCPU100からデータブロックが順に送られる。ディスコネクト・リコネクト機能により各磁気ディスク装置がI/Oバスインタフェース150と順に接続されて、データブロック0、1、2、3がid番号0番、データブロック4、5、6、7がid番号1番、データブロック8、9、10、11がid番号2番、データブロック12、13、14、15がid番号3番、データブロック16、17、18、19がid番号0番... の磁気ディスク装置へとそれぞれ格納される。なお、格納する磁気ディスク装置の順序は必ずしもid番号の順である必要はなく、0番以外の磁気ディスク装置から開始することも可能である。また、id番号の順序を昇順、または降順以外とすることも可能である。

【0157】図21はデータブロック4個を単位としてファイルを先頭から順に読み出す場合を示している。I/Oバスインタフェース150を介して、ホストCPU100へデータブロックが順に送られる。図20に示したようにデータブロックは格納されており、ディスコネクト・リコネクト機能により各磁気ディスク装置がI/Oバスインタフェース150と順に接続されて、データブロック0、1、2、3がid番号0番、データブロック4、5、6、7がid番号1番、データブロック8、9、10、11がid番号2番、データブロック12、13、14、15がid番号3番、データブロック16、17、18、19がid番号0番... の磁気ディスク装置からとそれぞれ読み出される。なお、格納する磁気ディスク装置の順序は必ずしもid番号の順である必要はなく、0番以外の磁気ディスク装置から開始することも可能である。また、id番号の順序を昇順、または降順以外とすることも可能である。

【0158】なお、ストライピングの単位とするデータブロック数は、4個以外の任意の個数を取りうることは

10

20

30

40

50

言うまでもない。

【0159】図22～33を用いて、論理バーチャルアレイドisk装置として使用する論理disk装置の組合せについて説明する。

【0160】本実施例では、disk管理テーブル210で磁気disk装置のid番号または名称と、パーティション番号とを指定して使用する論理disk装置を定義する。図7の説明で述べたように、複数の論理disk装置に一連の識別子vaXY0、vaXY

1、... を付けることで、論理バーチャルアレイドisk装置として利用する論理disk装置の組合せの定義が可能である。識別子の添字の意味は、図7のdisk管理テーブルの説明で述べたとおりである。本実施例ではサブファイルの物理的な配置の自由度が非常に高く、以下のような構成が可能である。なお、例えば図7のdisk装置名称hd4、hd5のdisk装置のように、論理バーチャルアレイドisk装置として定義していない領域は、特に断わりがなくとも通常の論理disk装置として利用可能である。

【0161】図22はid番号0、1、2、3の4台の磁気disk装置のすべてのパーティションを論理バーチャルアレイドisk装置として用いる場合である。

【0162】これに対応するdisk管理テーブル210の設定例を図23に示す。4台の磁気disk装置をバーチャルアレイドisk装置va0として定義し、全体をひとつの論理バーチャルアレイドisk装置va0として用いる場合に相当する。id番号4、5、6の3台の磁気disk装置は、通常の磁気disk装置として用いる。

【0163】図24はid番号0、1、2、3の4台の磁気disk装置の特定のパーティションを用いる場合である。

【0164】これに対応するdisk管理テーブルの設定例を図25に示す。図23と同様に定義したバーチャルアレイドisk装置va0の第0パーティションを論理バーチャルアレイドisk装置va00として用いる場合である。id番号4、5、6の3台の磁気disk装置は、通常の磁気disk装置として用いる。

【0165】図26はid番号0、1、2、3の4台の磁気disk装置の各3つのパーティションを用いて、論理バーチャルアレイドisk装置を3組定義した場合である。

【0166】これに対応するdisk管理テーブル210の設定例を図27に示す。図23と同様に定義したバーチャルアレイドisk装置va0の第0、第1、第2パーティションを、それぞれ論理バーチャルアレイドisk装置va00、va01、va02として用いる場合である。id番号4、5、6の3台の磁気disk装置は、通常の磁気disk装置として用いる。

【0167】図28はid番号0、1、2、3の4台の

磁気disk装置の特定のパーティション、およびid番号4、5の2台の磁気disk装置の特定のパーティションを用い、論理バーチャルアレイドisk装置を2組定義した場合である。

【0168】これに対応するdisk管理テーブル210の設定例を図29に示す。id番号0、1、2、3の4台の磁気disk装置をバーチャルアレイドisk装置va0、id番号4、5の2台の磁気disk装置をバーチャルアレイドisk装置va1としてそれぞれ定義し、各々の第0パーティションをそれぞれ論理バーチャルアレイドisk装置va00、va10として用いる場合である。id番号6の磁気disk装置は、通常の磁気disk装置として用いる。

【0169】図30はid番号0、1、2、3の4台の磁気disk装置の各3つのパーティション、およびid番号4、5の2台の磁気disk装置の各3つのパーティションを用い、論理バーチャルアレイドisk装置を6組定義した場合である。

【0170】これに対応するdisk管理テーブル210の設定例を図31に示す。図29と同様に定義したバーチャルアレイドisk装置va0、va1の第0、第1、第2パーティションを、それぞれ論理バーチャルアレイドisk装置va00、va01、va02、およびva10、va11、va12として用いる場合である。id番号6の磁気disk装置は通常の磁気disk装置として用いる。

【0171】図32はid番号0、2の2台の磁気disk装置の各3つのパーティション、id番号4、6の2台の磁気disk装置の各2つのパーティション、およびid番号1、3の2台の磁気disk装置の各1つのパーティションを用い、論理バーチャルアレイドisk装置を3組定義した場合である。

【0172】これに対応するdisk管理テーブル210の設定例を図33に示す。id番号0、2、4、6の4台の磁気disk装置をバーチャルアレイドisk装置va0、id番号0、1、2、3の4台の磁気disk装置をバーチャルアレイドisk装置va1としてそれぞれ定義し、va0の第0、第2パーティションをそれぞれ論理バーチャルアレイドisk装置va00、va02として、va1の第1パーティションを論理バーチャルアレイドisk装置va11として用いる場合である。id番号0、2の2台の磁気disk装置はva0とva1とに重複して定義されることになるが、利用するパーティションが重ならないようにすれば問題は生じない。なお、id番号5の磁気disk装置は、通常の磁気disk装置として用いる。以上の各構成例において、磁気disk装置の台数及びid番号は、上記以外の組合せでも実現可能である。

【0173】図34～39は本実施例においてミラーモードを実現する例である。すなわち、論理バーチャルア

10

20

30

40

50

レイディスク装置としてプライマリ（正）とバックアップ（副）を一組にして定義し、同一ファイルを両方に格納するものである。これにより、プライマリに障害が発生した場合には、バックアップを代わりに使用することにより信頼性向上を図ることができる。

【0174】ファイル書き込みの場合には、プライマリとバックアップの両方に書き込む。

【0175】まずプライマリとして定義した論理バーチャルレイディスク装置に対して、非同期書き込みによりデータブロックを書き込む。引き続き、バックアップとして定義した論理バーチャルレイディスク装置に対しても、やはり非同期書き込みによりプライマリと同様のデータブロックを書き込む。バックアップへの書き込み命令を発行した後、次のデータブロックの書き込み処理に移り、プライマリおよびバックアップに対して上記と同様の手順によりデータブロックを書き込む。

【0176】なお、プライマリとバックアップへの書き込み方に関しては、各論理バーチャルレイディスク装置を構成する対応するディスク装置ごとに、すなわちhd0とhd2、hd1とhd3というように書き込むことも、その逆にhd0とhd3、hd1とhd2というように書き込むことも可能である。これは、ディスク管理テーブルに設定された識別子v a X Y ZのZの値をキーとして対応を取ることににより行う。具体的な例は図35の説明で述べる。

【0177】これにより、ユーザにはバックアップへの書き込み時間をほとんど感じさせずに、ミラーモードを実現することができる。

【0178】ファイル読み出しの場合には、プライマリのみから読み出しを行う。すなわち、ミラーモード指定を行っていない論理バーチャルレイディスク装置に格納されたファイルを読み出すのと同様の方法で、プライマリからファイルの読み出しを行う。しかし、プライマリに障害が発生すると、バックアップに格納されたファイルを代わりに使用する。

【0179】これにより、磁気ディスク装置の信頼性を向上することができる。また、通常の利用状態ではプライマリのみからファイルを読み出すので、バックアップが存在することによるペナルティが発生することなく、高速にファイルの読み出しが行える。

【0180】以下、図を用いて、ミラーモードでの論理バーチャルレイディスク装置として使用する論理ディスク装置の組合せについて説明する。ミラーモードでない論理バーチャルレイディスク装置の定義の場合と同様に、ディスク管理テーブル210により使用する論理ディスク装置に一連の識別子を付けることで定義を行う。さらに、プライマリであるのかバックアップであるのかということと、その組合せを拡張子により示す。例えば、v a X Y Z _ p 0は識別子v a X Y Zの論理ディスク装置がシステム中でミラーモードのプライマリ論理

バーチャルレイディスク装置の構成要素として定義されており、バックアップ論理バーチャルレイディスク装置の構成要素として定義されるv a X Y Z _ b 0の論理ディスク装置と対をなすことを示す。なお、論理バーチャルレイディスク装置として定義していない領域は、特に断わりがなくとも通常の論理ディスク装置として利用可能である。

【0181】図34はid番号0、1の2台の磁気ディスク装置の各1つのパーティションを用いてプライマリとし、id番号2、3の2台の磁気ディスク装置の各1つのパーティションを用いてバックアップとして、ミラーモードの論理バーチャルレイディスク装置を1組定義した場合である。

【0182】これに対応するディスク管理テーブル210の設定例を図35に示す。id番号0、1の2台の磁気ディスク装置をバーチャルレイディスク装置v a 0、id番号2、3の2台の磁気ディスク装置をバーチャルレイディスク装置v a 1としてそれぞれ定義し、各々の第0パーティションをそれぞれ論理バーチャルレイディスク装置v a 0 0 _ p 0（プライマリ）、v a 1 0 _ b 0（バックアップ）として用いる場合である。本図の例では、プライマリを構成する論理ディスク装置v a 0 0 0 - p 0、v a 0 0 1 - p 0に、バックアップを構成する論理ディスク装置v a 1 0 0 - b 0、v a 1 0 1 - b 0がそれぞれ対応する。id番号4、5、6の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【0183】図36はid番号0、1、2、3の4台の磁気ディスク装置の各2つのパーティションを用いて、第1パーティションをプライマリとし、第2パーティションをバックアップとして、ミラーモードの論理バーチャルレイディスク装置を1組定義した場合である。

【0184】これに対応するディスク管理テーブル210の設定例を図37に示す。id番号0、1、2、3の4台の磁気ディスク装置をバーチャルレイディスク装置v a 1として定義し、第1、第2パーティションをそれぞれ論理バーチャルレイディスク装置v a 1 1 _ p 1（プライマリ）、v a 1 2 _ b 1（バックアップ）として用いる場合である。id番号4、5、6の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【0185】図38はid番号0、1、2、3の4台の磁気ディスク装置のひとつのパーティションを用いてプライマリとし、id番号3、4、5、6の4台の磁気ディスク装置のひとつのパーティションを用いてバックアップとして、ミラーモードの論理バーチャルレイディスク装置を1組定義した場合である。

【0186】これに対応するディスク管理テーブル210の設定例を図39に示す。id番号0、1、2、3の4台の磁気ディスク装置をバーチャルレイディスク装置v a 2、id番号3、4、5、6の4台の磁気ディ

10

20

30

40

50

ク装置をバーチャルアレイディスク装置 $v a 3$ としてそれぞれ定義し、各々の第3、第4パーティションをそれぞれ論理バーチャルアレイディスク装置 $v a 2 3_p 2$ (プライマリ)、 $v a 3 4_b 2$ (バックアップ) として用いる場合である。

【0187】図40は本発明の第3の実施例である。図1に示した第1の実施例における $i d$ 番号0、1、2、3の磁気ディスク装置170-0~170-3をアレイディスク装置300-0~300-3に置き換えたものである。アレイディスク装置は磁気ディスク装置に比べて装置単体でのファイル転送性能が高いので、本実施例は第1の実施例に比べてより一層ファイルアクセス高速化の効果が得られる。

【0188】なお、動作や制御方式は図1の第1の実施例と全く同様である。

【0189】これに対応する、アレイディスク装置300-0~300-3を含むバーチャルアレイディスクのディスク管理テーブルの設定例を図41に示す。 $i d$ 番号0、1、2、3の4台がアレイディスク装置となっている。ここでは4台のアレイディスク装置 $a d 0$ 、 $a d 1$ 、 $a d 2$ 、および $a d 3$ でバーチャルアレイディスク装置 $v a 0$ を構成している。

【0190】また本実施例においても、論理バーチャルアレイディスク装置 $v a 0 1_p 0$ と $v a 0 2_b 0$ 、 $v a 2 1_p 1$ と $v a 2 2_b 1$ のようにミラーモードに対応することも可能である。

【0191】これまでの実施例では7台の磁気ディスク装置が接続された場合を例に取り説明したが、磁気ディスク装置の台数を2台、3台、4台、...と順次増やして行くことも可能である。ここで、例えば4台の磁気ディスク装置を5台に増やした場合、4台のディスクに格納されていたファイルを5台の磁気ディスク装置に再配置することが必要となる。

【0192】本発明によれば、磁気ディスク装置の増設後に新たなパーティションへファイルをコピーすることにより、ファイルのストライピング数を自動的に増加させて行くことも可能となる。すなわち、バーチャルアレイディスク装置に格納されたファイルを読み書きする場合、ユーザはファイルの物理的な分割を意識することなく処理を行うことができるため、コピー元からコピー先へファイルをコピーするだけで、ユーザがこれを特に意識することなく、ストライピング数の変換を行うことが可能である。この場合について以下に説明する。

【0193】図42にディスク管理テーブル210の設定例を、図43にディスク増設時のファイルコピーの制御フローをそれぞれ示す。図42において、磁気ディスク装置の $i d$ 番号0、1、2、3の第0パーティションにより構成されている論理バーチャルアレイディスク装置 $v a 0 0$ はコピー元であり、磁気ディスク装置の $i d$ 番号0、1、2、3、4の第1パーティションにより構

成されている論理バーチャルアレイディスク装置 $v a 1 1$ はコピー先である。

【0194】図43に示す制御フローに従って、 $v a 0 0$ から $v a 1 1$ へファイルを固定長のデータに分割してコピーする。

【0195】(1) コピー先の論理バーチャルアレイディスク装置 $v a 1 1$ をディスク管理テーブル210に定義する。

【0196】(2) コピー元の論理バーチャルアレイディスク装置 $v a 0 0$ から、図14のファイル読み出しの制御フローによりファイルを分割した固定長のデータを読み込み、これをコピー先の論理バーチャルアレイディスク装置 $v a 1 1$ へ、図15のファイル書き込みの制御フローにより書き込む。

【0197】(3) コピー終了か判定し、まだ残りがあれば(2)に戻って繰り返す。コピー終了であれば

(4) へ進む。

【0198】(4) ディスク管理テーブル210上のコピー元の論理バーチャルアレイディスク装置 $v a 0 0$ を無効化する。

【0199】以上により、磁気ディスク装置の増設に伴うストライピング数の変更が容易に可能となる。この際に、読み出しおよび書き込みの単位とする固定長のデータとしては、任意のブロック数を定義することができる。

【0200】図44は複数のI/Oバスを有する本発明の一実施例である。図1の第1の実施例の構成ではI/Oバスは1本であったが、図44では4本のI/Oバスを有する構成となっている。したがって、I/Oバスインタフェース150、151、152、153とI/Oバス160、161、162、163、さらに各I/Oバスごとの7台の磁気ディスク装置170-0~170-6、171-0~171-6、172-0~172-6、173-0~173-6からなる4組のディスクサブシステムを設けている。アクセス制御プログラムが複数のI/Oバスへのスケジューリングを行う点を除けば、各ディスクサブシステムの制御方法は前述の各実施例のものをそのまま適用することが可能である。例えば、各I/Oバスごとに第1の実施例のような多重アクセス制御を行い、さらにI/Oバス間での1階層上での多重アクセス制御を行うということが可能である。また、各I/Oバスごとの多重アクセス制御を並列に行うことも可能である。

【0201】各I/Oバス160、161、162、163と共通データバス101間でのデータ転送において、共通データバスの速度がI/Oバスの速度の4倍よりも大きく、なおかつ上述のような制御方式を用いることによりデータ転送の相手が特定のI/Oバスに集中することがないような場合には、本図の実施例は最大のデータ転送速度を達成することができる。その場合、I/O

10

20

30

40

50

Ｏバスが１本の第１の実施例に比較して本図の実施例では４倍のデータ転送速度となる。したがって、本実施例によりバーチャルアレイディスクの高速性をさらに高めることが可能となる。なお、本図ではＩ／Ｏバスが４本の場合について示したが、もちろん４本以外の構成を取ることも可能である。

【０２０２】また、磁気ディスク装置を増設してシステムを拡張する場合には、各Ｉ／Ｏバス内での磁気ディスク装置の増設と、Ｉ／Ｏバス単位での増設が可能である。このため、１スタックごとに磁気ディスク装置の増設を行わなければならないアレイディスク装置に比べ、よりフレキシブルにシステムの拡張に対応することができるといふ長所を有する。

【０２０３】この様に本実施例によれば、バーチャルアレイディスク装置を用いたファイルアクセスの一層の高速化と、柔軟なシステムの拡張性を同時に実現することができるという効果がある。

【０２０４】図４０の第３の実施例に示したように、図４４の実施例の磁気ディスク装置をアレイディスク装置に置き換えた構成を取ることも可能である。この実施例を図４５に示す。図４５では、第０番のＩ／Ｏバスの磁気ディスク装置がアレイディスク装置３００-０～３００-６となっている。これにより、より一層のファイルアクセス高速化を図ることが可能となる。この際、原理的にはアレイディスク装置は任意の磁気ディスク装置と入れ替えることが可能であり、本図に示した以外の構成も容易に実現することが可能である。

【０２０５】なお、上記図４４、および図４５に示した複数のＩ／Ｏバスを有する実施例においても、１本のＩ／Ｏバスを有する実施例の説明中で述べた本発明の種々の機能を実現することが可能なことは明らかである。

【０２０６】本発明によれば、ディスコネクト・リコネクト機能を備えたＩ／Ｏバスを有する計算機システムにおいて、該Ｉ／Ｏバスに接続された複数のディスク装置にファイルを分割格納してソフトウェアで多重・並列アクセス制御することで、高価なハードウェアを用いることなく、アレイディスク装置に匹敵するほどの高速なファイルアクセスを実現することができるという効果がある。しかも、ユーザはあたかも１台のディスク装置を使っているかのごとく、複数のディスクに分割されたファイルをアクセスすることができるようになる。これにより、従来よりも非常に低コストで高速ファイルアクセス可能な計算機システムを実現することができるという効果が得られる。

【０２０７】また、特殊なハードウェアを用いないので、ディスク装置の増設の際には１台ずつ増設していくことができ、柔軟なシステム拡張が可能であるという効果もある。

【０２０８】

【発明の効果】本発明によれば、高速なデータアクセス

可能な計算機システムを実現することができるという効果が得られる。

【図面の簡単な説明】

【図１】第１の実施例を示す図

【図２】第１の実施例におけるファイル読み出しのタイムチャート

【図３】第１の実施例におけるファイル読み出しのフローチャート

【図４】第１の実施例におけるファイル書き込みのタイムチャート

【図５】第１の実施例におけるファイル書き込みのフローチャート

【図６】第２の実施例を示す図

【図７】ディスク管理テーブルを示す図

【図８】ファイル管理テーブルを示す図

【図９】ファイル記述子対応テーブルを示す図

【図１０】ファイルアクセスの全体フローチャート

【図１１】mount処理の制御フローチャート

【図１２】ディレクトリ構造を示す図

【図１３】open処理の制御フローチャート

【図１４】read処理の制御フローチャート

【図１５】write処理の制御フローチャート

【図１６】close処理の制御フローチャート

【図１７】umount処理の制御フローチャート

【図１８】データブロック１個を単位とするストライピングを示す図

【図１９】データブロック１個を単位とするストライピングを示す図

【図２０】データブロック４個を単位とするストライピングを示す図

【図２１】データブロック４個を単位とするストライピングを示す図

【図２２】バーチャルアレイディスクの使用領域を示す図

【図２３】ディスク管理テーブルの設定例を示す図

【図２４】バーチャルアレイディスクの使用領域を示す図

【図２５】ディスク管理テーブルの設定例を示す図

【図２６】バーチャルアレイディスクの使用領域を示す図

【図２７】ディスク管理テーブルの設定例を示す図

【図２８】バーチャルアレイディスクの使用領域を示す図

【図２９】ディスク管理テーブルの設定例を示す図

【図３０】バーチャルアレイディスクの使用領域を示す図

【図３１】ディスク管理テーブルの設定例を示す図

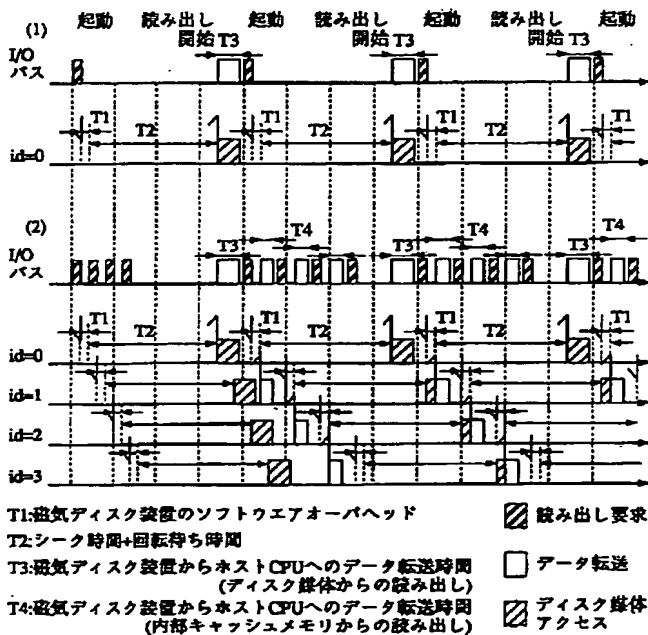
【図３２】バーチャルアレイディスクの使用領域を示す図

【図３３】ディスク管理テーブルの設定例を示す図

- 【図34】ミラーモードの使用領域を示す図
 【図35】ミラーモードでのディスク管理テーブルの設定例を示す図
 【図36】ミラーモードの使用領域を示す図
 【図37】ミラーモードでのディスク管理テーブルの設定例を示す図
 【図38】ミラーモードの使用領域を示す図
 【図39】ミラーモードでのディスク管理テーブルの設定例を示す図
 【図40】アレイディスクを含む構成の第3の実施例を示す図
 【図41】アレイディスクを含むディスク管理テーブルの設定例を示す図
 【図42】ディスク増設時のファイルコピーのディスク管理テーブルの設定例を示す図
 【図43】ディスク増設時のファイルコピーの制御フローチャート
 【図44】複数のI/Oバスを有する実施例を示す図
 【図45】複数のI/Oバスを有しアレイディスクを含む実施例を示す図
 【図46】アレイディスクの構成を示す図
 【図47】アレイディスクのタイムチャート
 【図48】本発明の原理図
 【図49】本発明におけるファイル読み出しのタイムチ

【図2】

図2



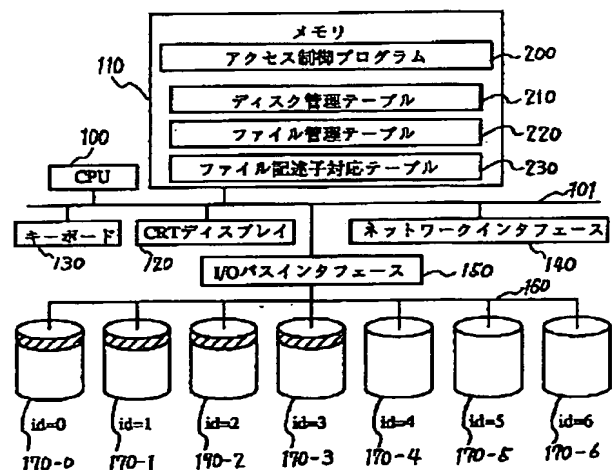
ヤート

【符号の説明】

10…ファイル、10-0～10-9…ファイルを構成するデータブロック、20-0～20-3…磁気ディスク装置の内部キャッシュメモリ、100…CPU、101…共通データバス、110…メモリ、120…CRTディスプレイ、130…キーボード、140…ネットワークインタフェース、150、151、152、153…I/Oバスインタフェース、160、161、162、163…I/Oバス、170-0～170-6…磁気ディスク装置、171-0～171-6…磁気ディスク装置、172-0～172-6…磁気ディスク装置、173-0～173-6…磁気ディスク装置、180-0～180-7…ディスクコネクタ/リコネクタ手段、181-7、182-7、183-7…ディスクコネクタ/リコネクタ手段、190…ワークステーション、191…LAN、200…アクセス制御プログラム、210…ディスク管理テーブル、220…ファイル管理テーブル、230…ファイル記述子対応テーブル、300、300-0～300-6…アレイディスク装置、310…FIFO0、311…FIFO1、312…FIFO2、313…FIFO3、320…内部バス、330…バッファ、340…SCSIバスインタフェース、400…ホストCPU装置。

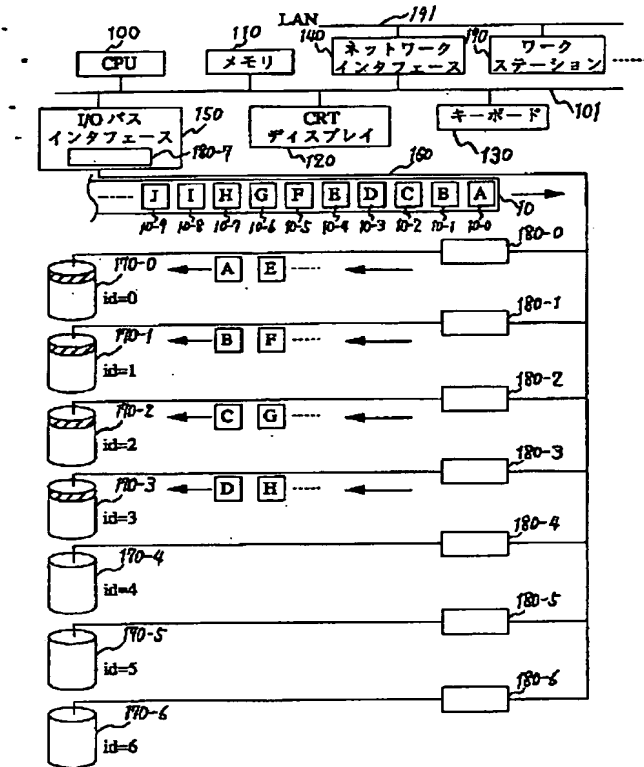
【図6】

図6



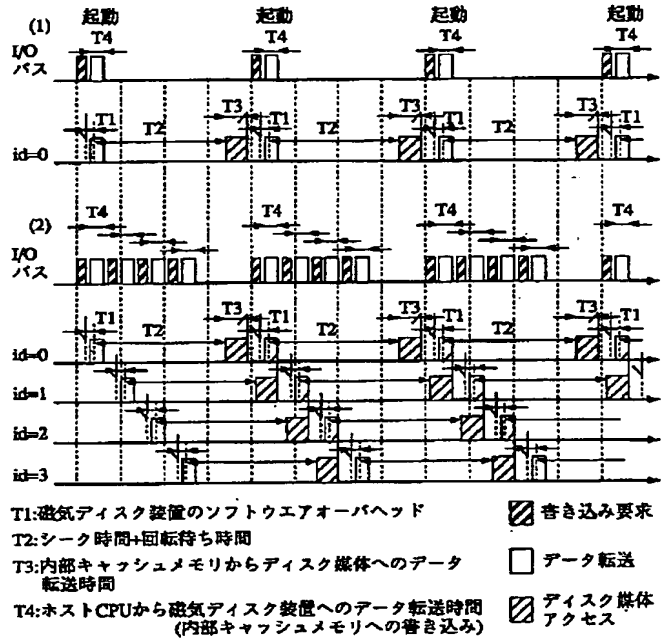
【図1】

図1



【図4】

図4



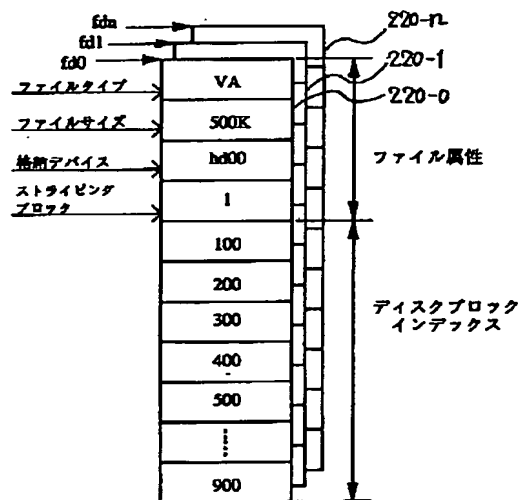
【図7】

図7

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va0	va0			
パーティション	0	va000	va001	va002	va003	va00 (論理VA装置)		hd60
	1					ディスク装置		hd61
	2	va020	va021	va022	va023	va02 (論理VA装置)		hd62
	3							hd63
	4					論理ディスク装置		hd64
	5					va0 (VA装置)		hd65

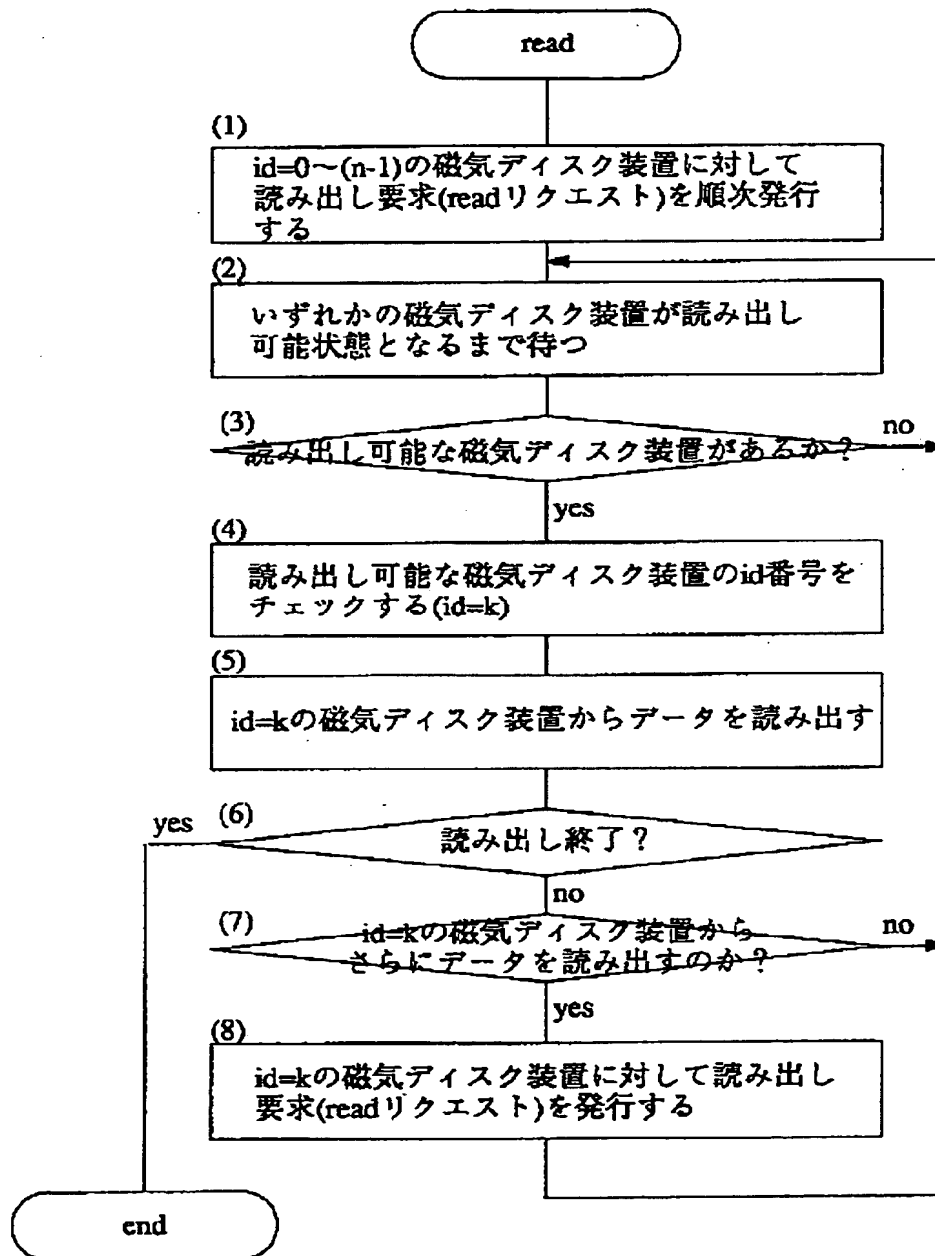
【図8】

図8



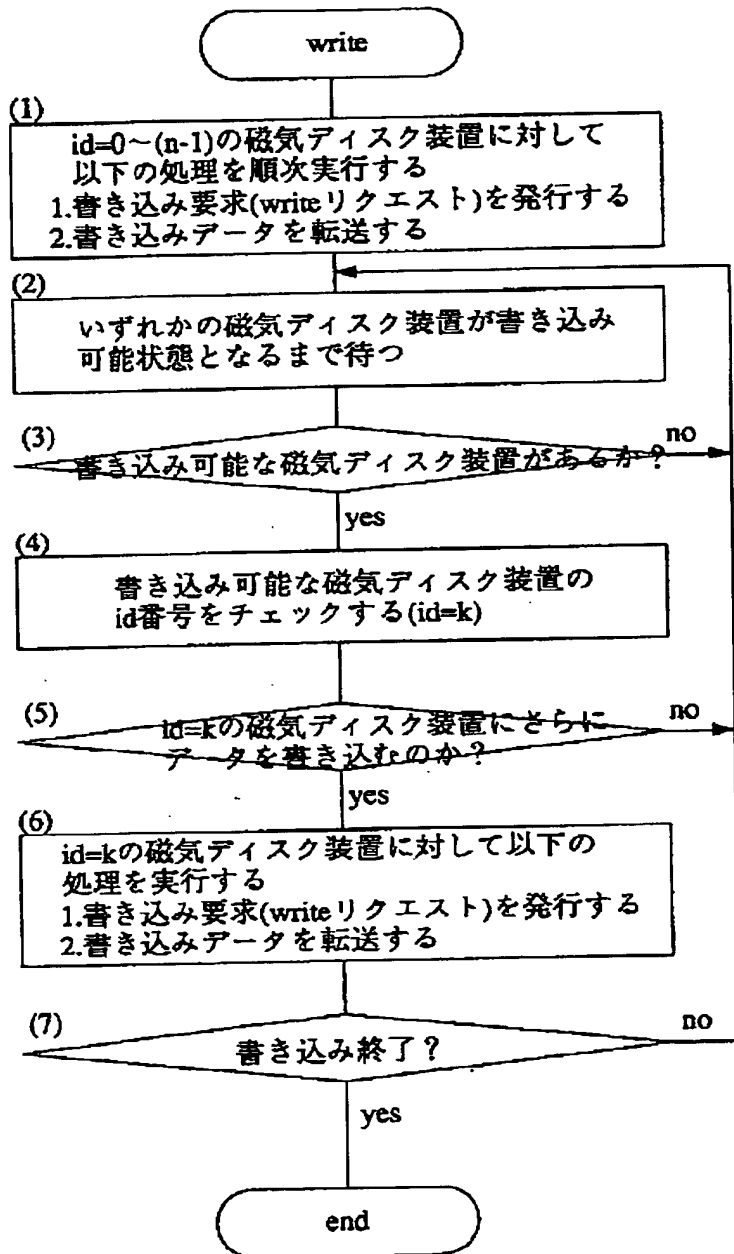
【図 3】

図 3



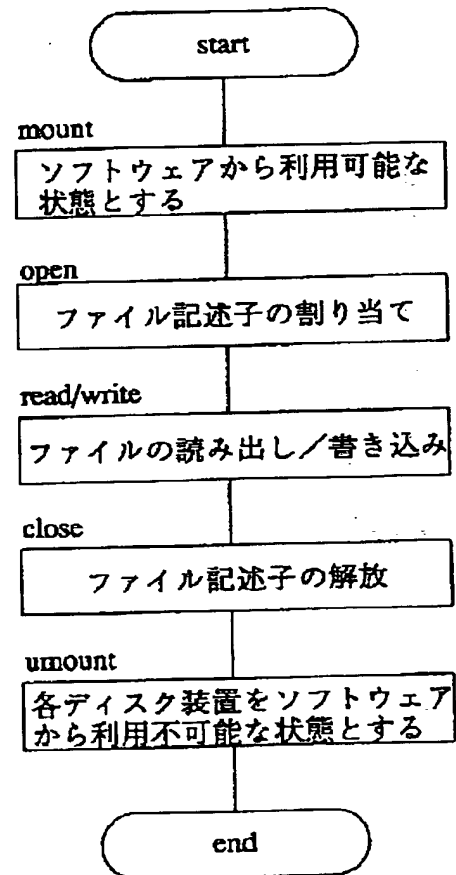
【図5】

図5



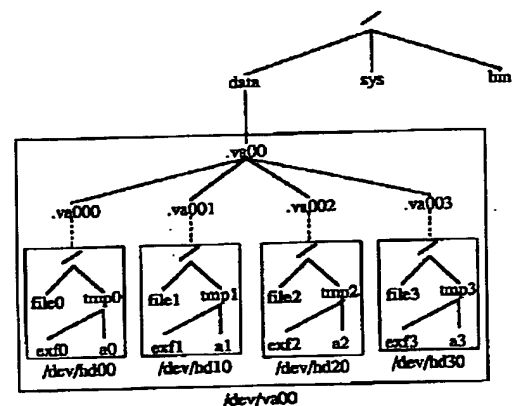
【図10】

図10



【図12】

図12



【図 9】

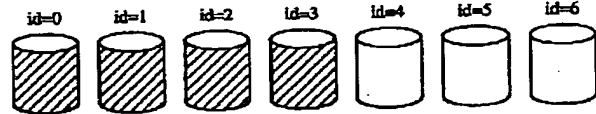
図 9

230

元ファイルの ファイル記述子	サブファイルのファイル記述子						
vfd	fd0	fd1	fd2	fd3	fd4	fd5	fd6
4	5	6	7	8			
20	21	22	23	24			

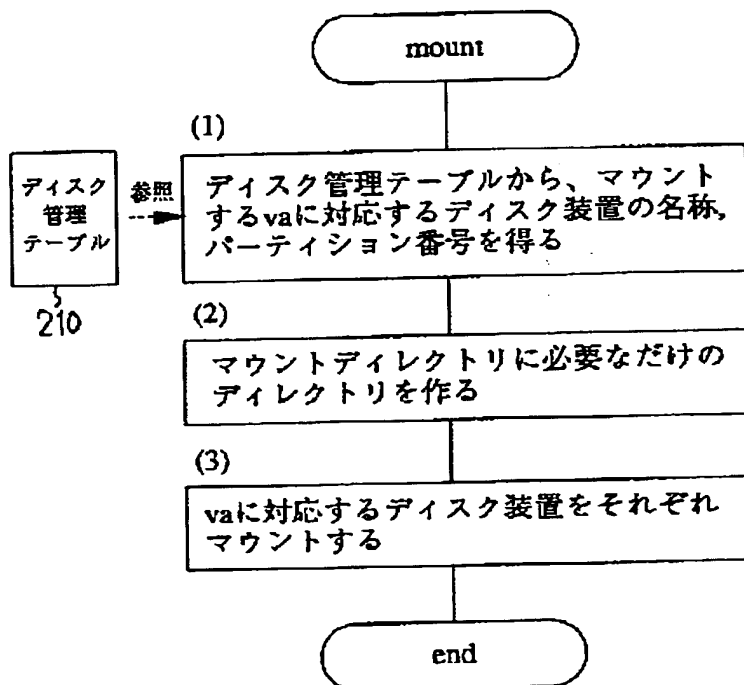
【図 2 2】

図 22



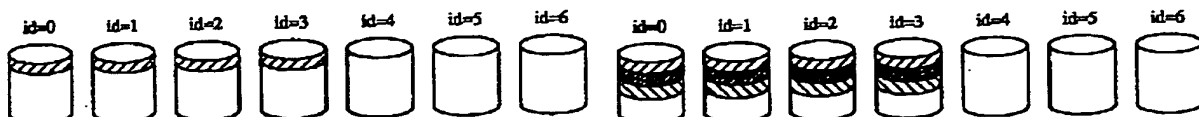
【図 1 1】

図 11



【図 2 4】

図 24

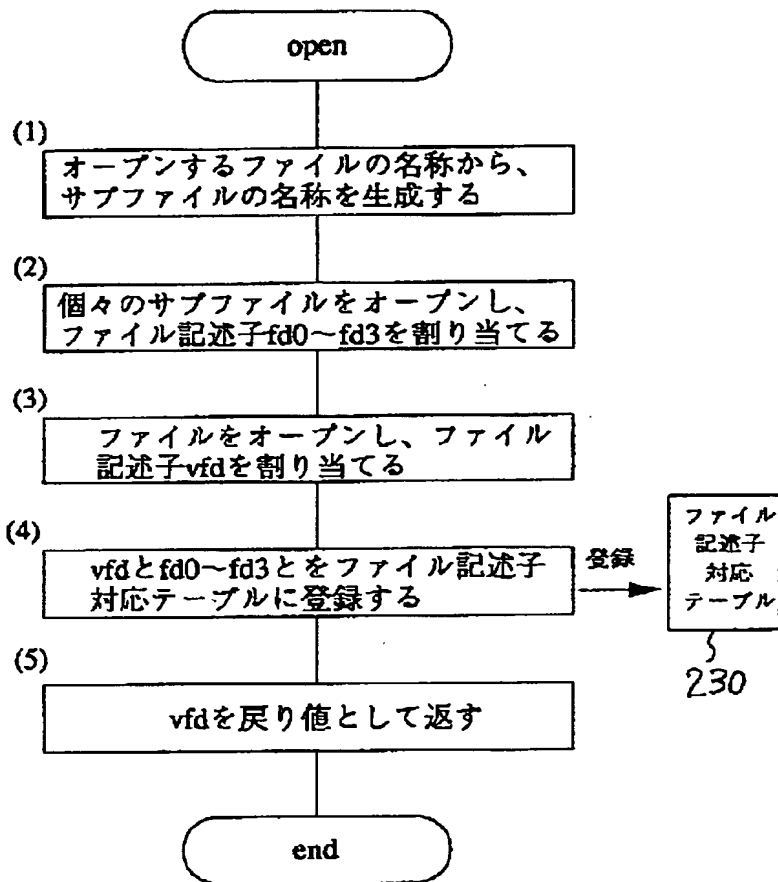


【図 2 6】

図 26

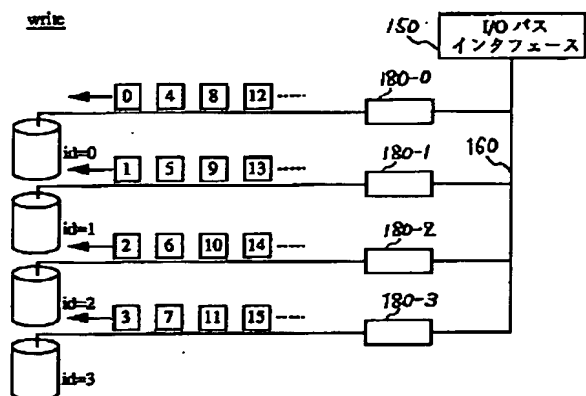
【図 1 3】

図 13



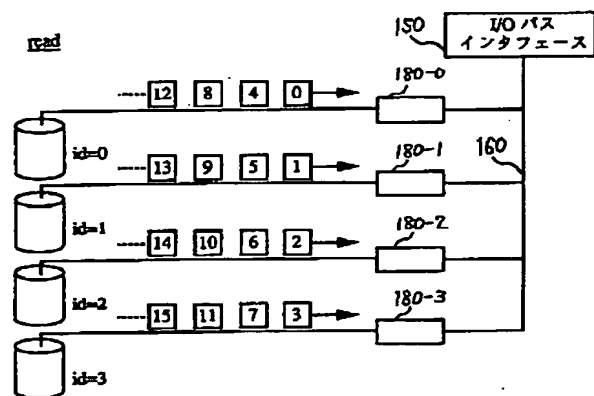
【図 1 8】

図18



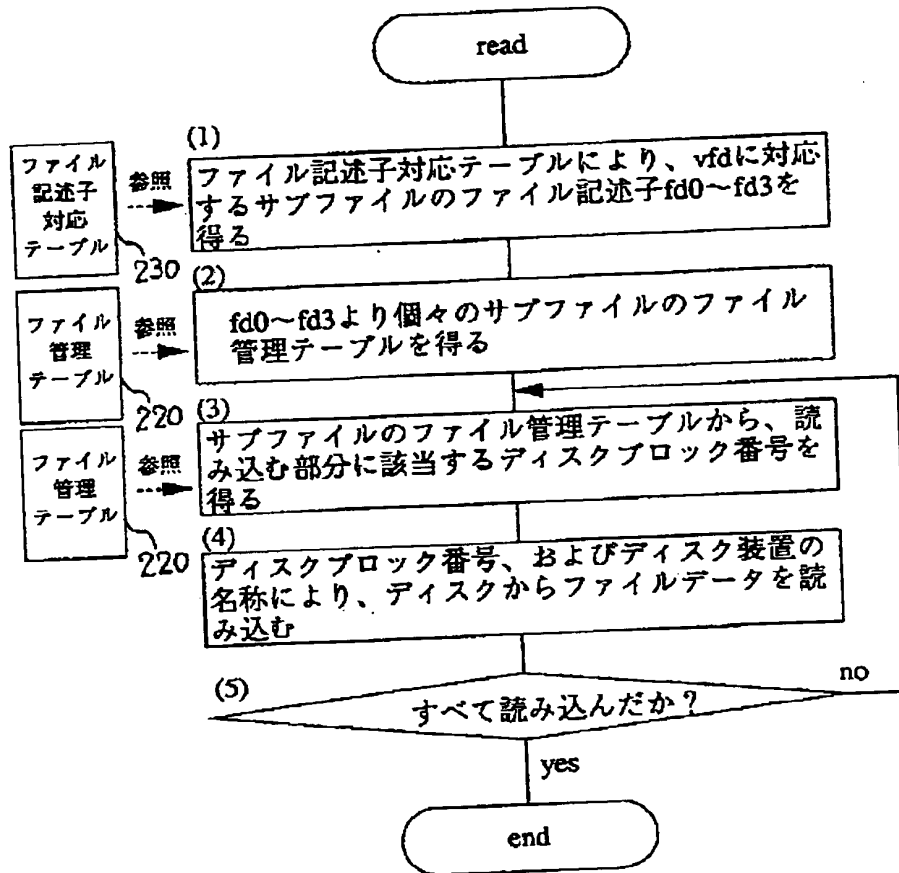
【図 1 9】

図19



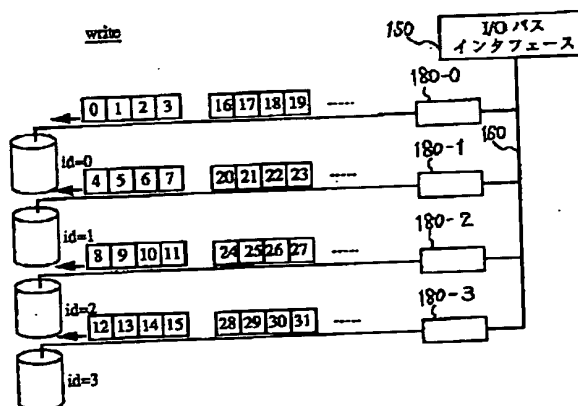
【図 1 4】

図 14



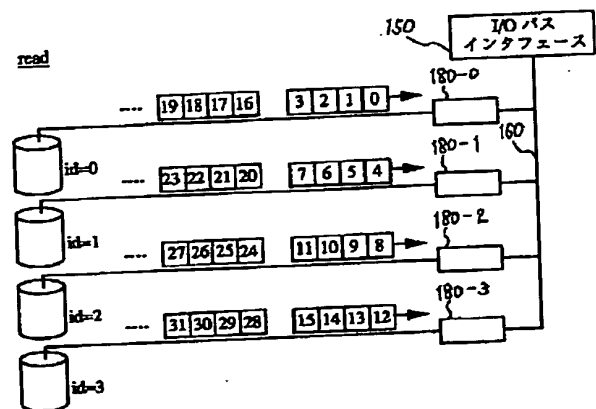
【図 2 0】

図 20



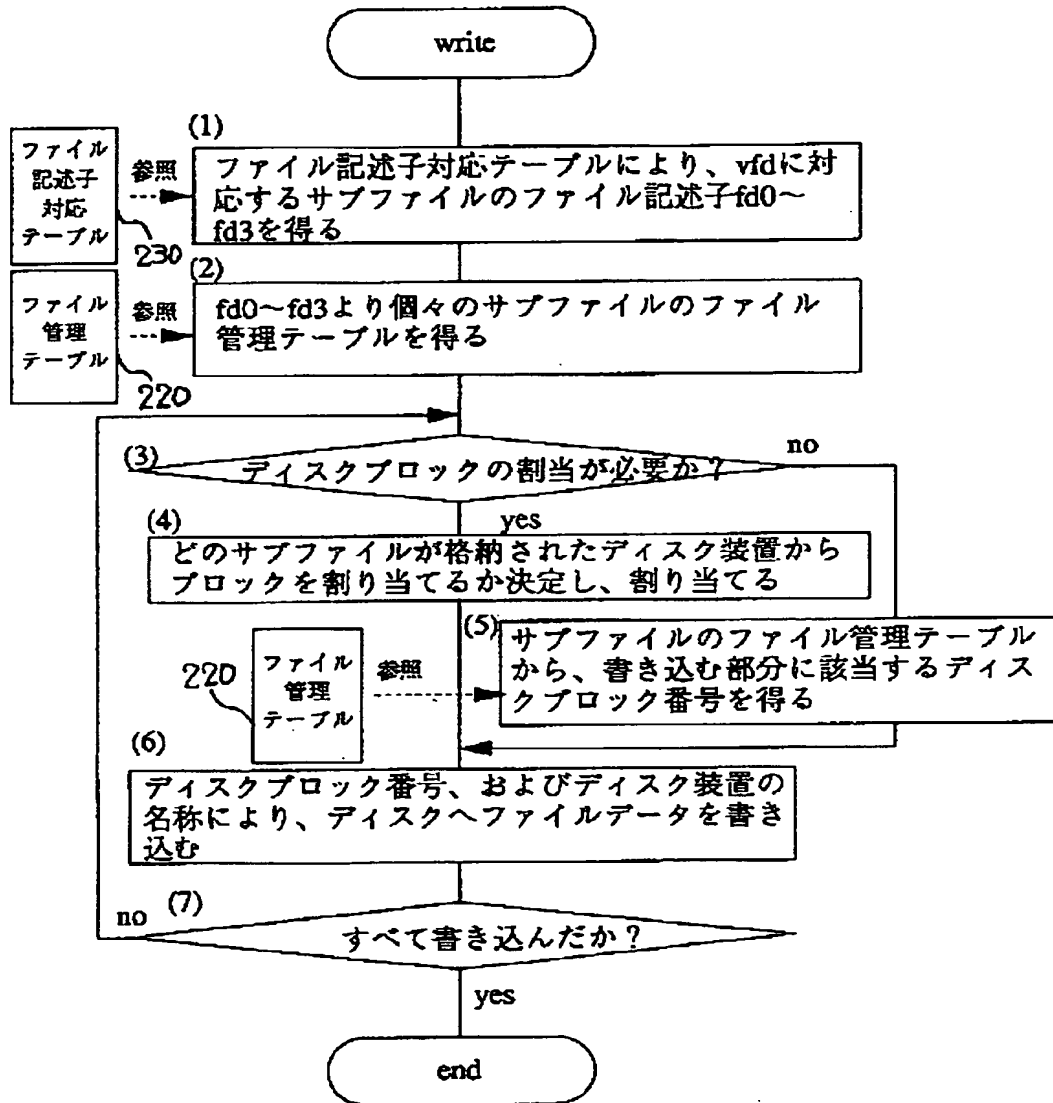
【図 2 1】

図 21



【図 15】

図 15



【図 28】

図 28



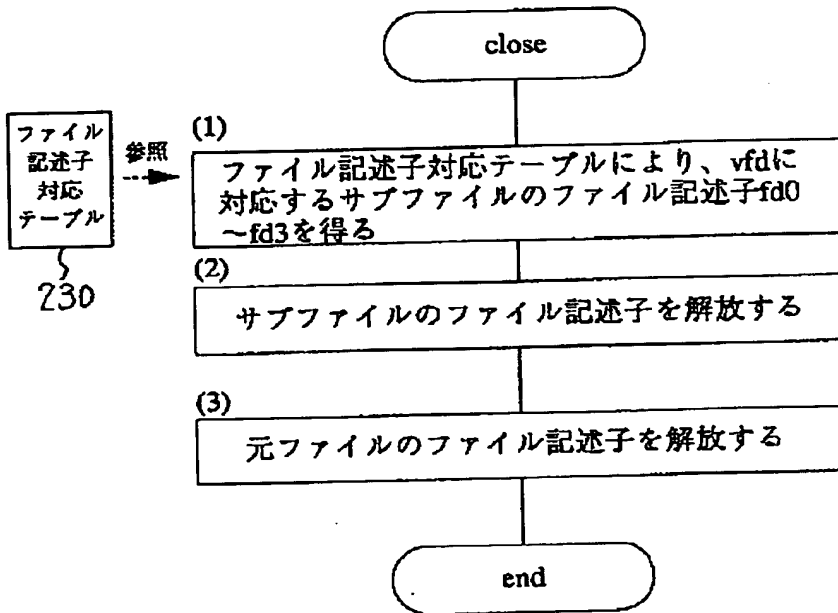
【図 30】

図 30



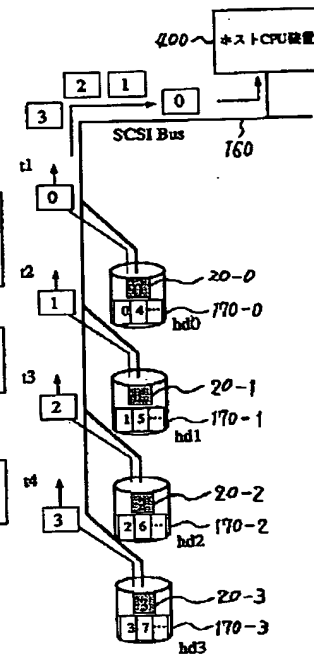
【図16】

図16



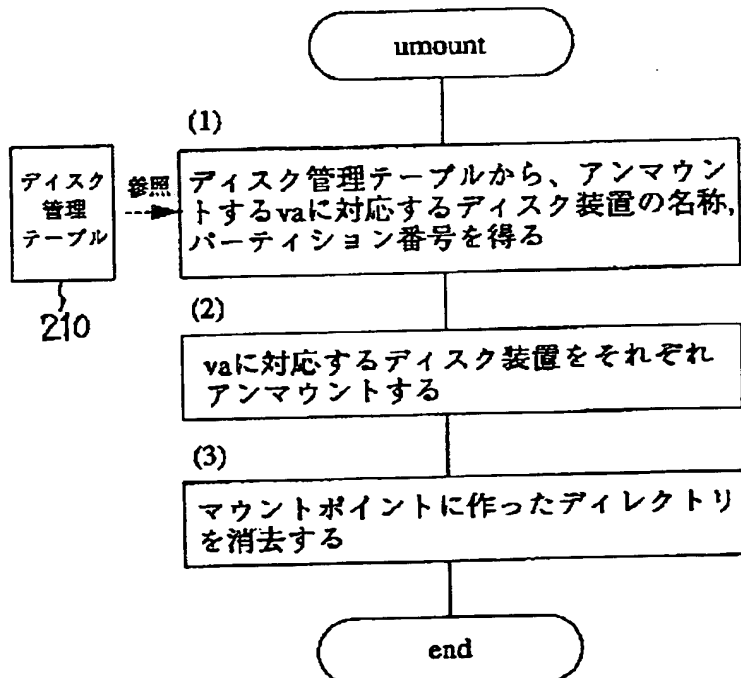
【図48】

図48



【図17】

図17



【図42】

図42

		磁気ディスク装置				
id番号		0	1	2	3	4
ディスク装置名称		hd0	hd1	hd2	hd3	hd4
VAディスク装置名称		va0, val	va0, val	va0, val	va0, val	val
パーティション	0	va000	va001	va002	va003	va00
	1	val10	val11	val12	val13	val14
	2					val1
	3					
	4					
	5					

【図23】

図23

210

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va0	va0			
パーティション	0	va000	va001	va002	va003	va00		
	1	va000	va001	va002	va003			
	2	va000	va001	va002	va003			
	3	va000	va001	va002	va003			
	4	va000	va001	va002	va003			
	5	va000	va001	va002	va003	va0		

【図25】

図25

210

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va0	va0			
パーティション	0	va000	va001	va002	va003	va00		
	1							
	2							
	3							
	4							
	5					va0		

【図27】

図27

210

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va0	va0			
パーティション	0	va000	va001	va002	va003	va00		
	1	va010	va011	va012	va013	va01		
	2	va020	va021	va022	va023	va02		
	3							
	4							
	5					va0		

【図29】

図29

210

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va0	va0	va1	va1	
パーティション	0	va000	va001	va002	va003	va100	va101	va10
	1				va00			
	2							
	3							
	4							
	5					va0	va1	

【図31】

図31

210

		ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va0	va0	va1	va1	
パーティション	0	va000	va001	va002	va003	va100	va101	va10
	1	va010	va011	va012	va013	va110	va111	va11
	2	va020	va021	va022	va023	va120	va121	va12
	3							
	4							
	5							

va0 va1

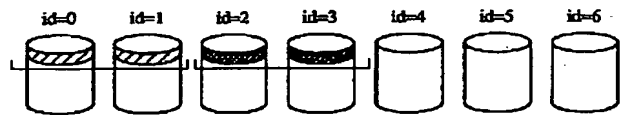
【図32】

図32



【図34】

図34



【図35】

図35

【図33】

図33

210

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0, va1	va1	va0, va1	va1	va0		va0
パーティション	0	va000		va001		va002		va003
	1	va110	va111	va112	va113	va11		
	2	va020		va021		va022		va023
	3							
	4							
	5							

va1 va0 va00 va02

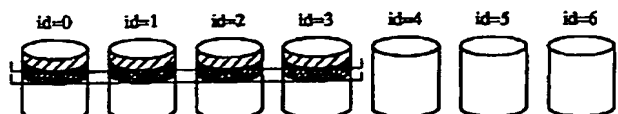
210

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va0	va0	va1	va1			
パーティション	0	va000-p0	va001-p0	va100-b0	va101-b0			
	1	va00-p0		va10-b0				
	2							
	3							
	4							
	5							

va0 va1

【図36】

図36



【図37】

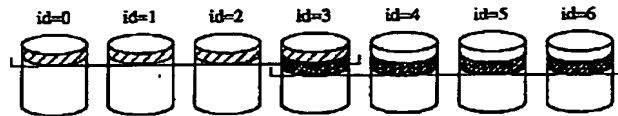
図37

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va1	va1	va1	va1			
パーティション	0							
	1	va110 -p1	va111 -p1	va112 -p1	va113 -p1	va11-p1		
	2	va120 -b1	va121 -b1	va122 -b1	va123 -b1	va12-b1		
	3							
	4							
	5							

va1

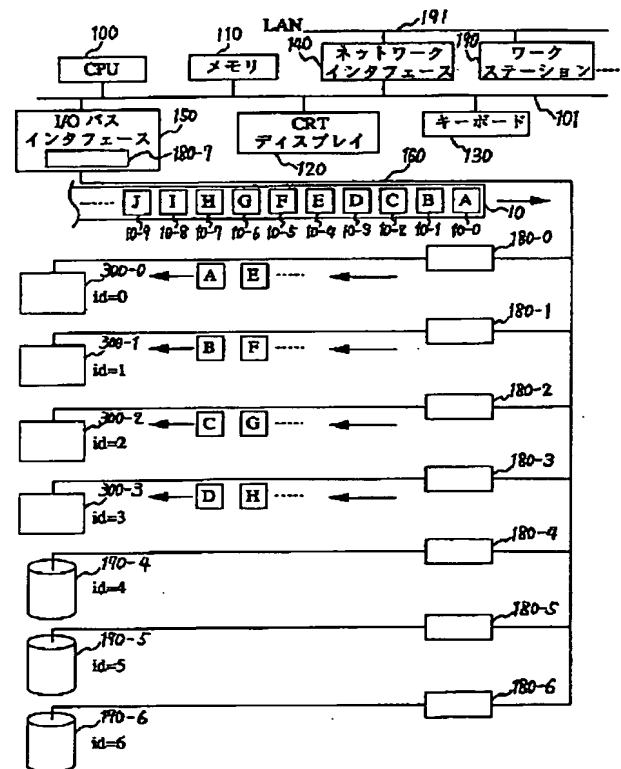
【図38】

図38



【図40】

図40



【図39】

図39

		磁気ディスク装置						
id番号		0	1	2	3	4	5	6
ディスク装置名称		hd0	hd1	hd2	hd3	hd4	hd5	hd6
VAディスク装置名称		va2	va2	va2	va2, va3	va3	va3	va3
パーティション	0							
	1							
	2							
	3	va230 -p2	va231 -p2	va232 -p2	va233 -p2	va23-p2		
	4				va340 -b2	va341 -b2	va342 -b2	va343 -b2
	5							

va2

va3

va34-b2

【图 4 4】

图 44

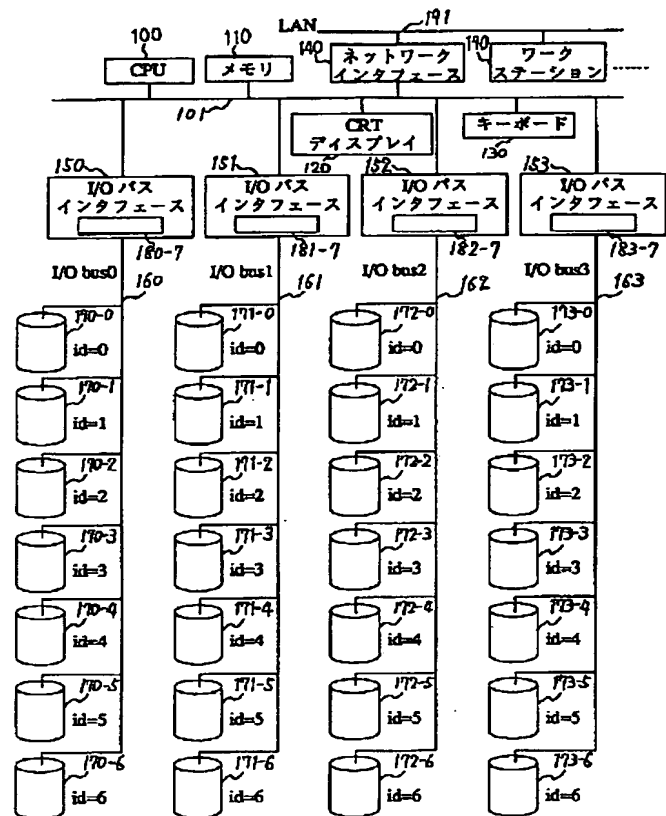
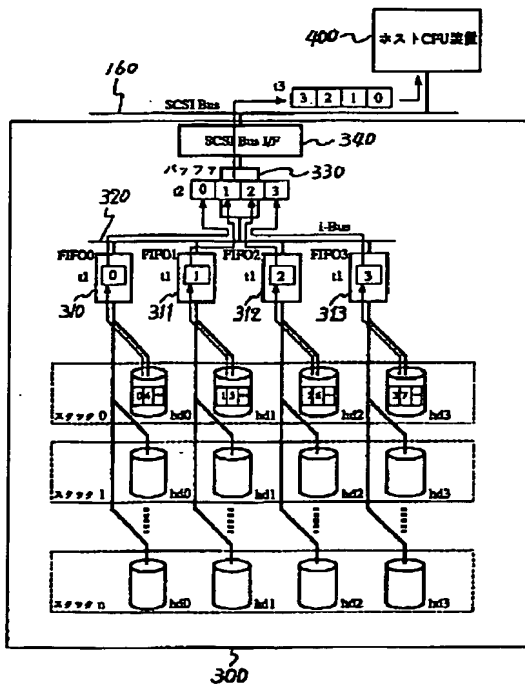
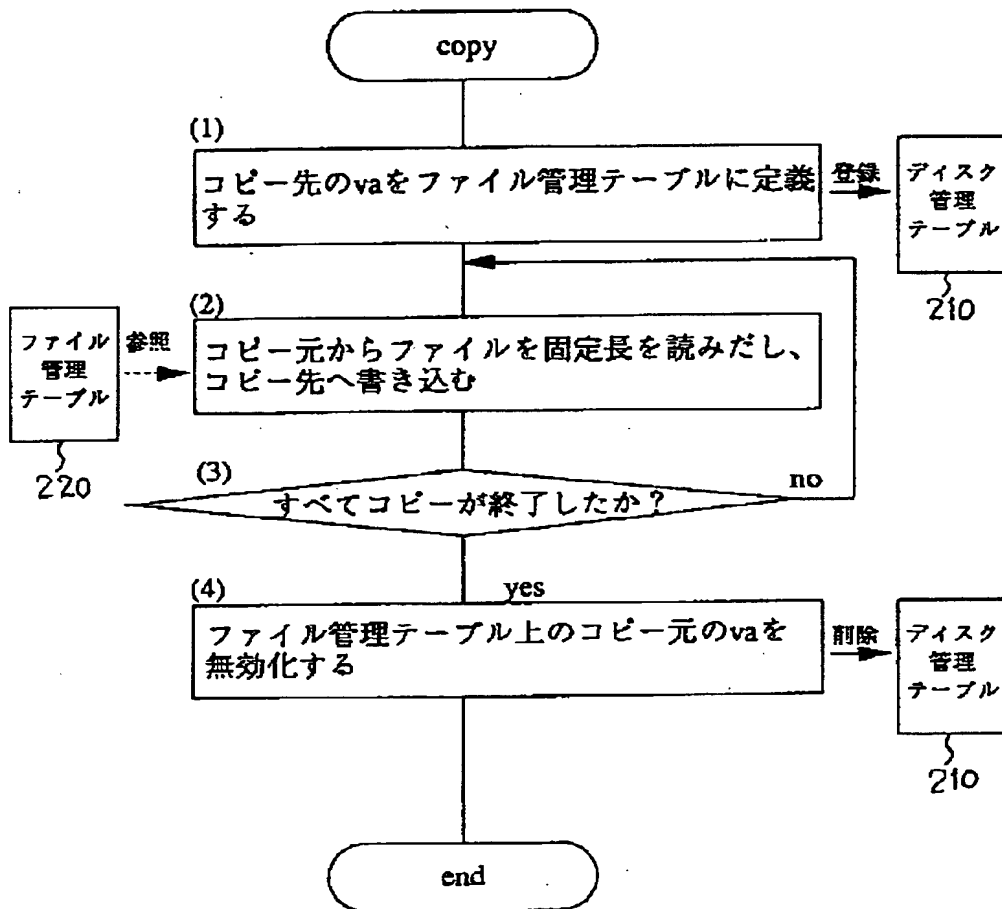


圖 46



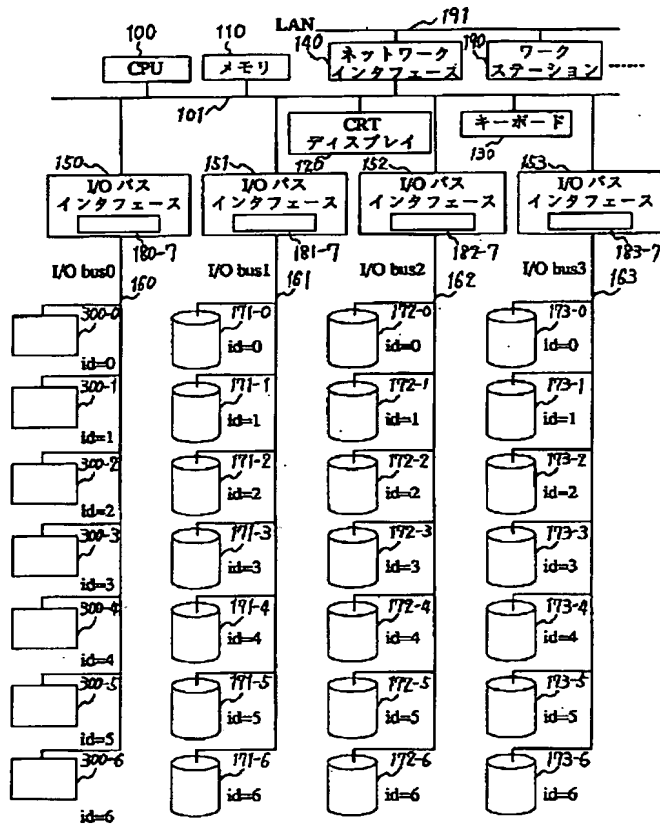
【図43】

図43



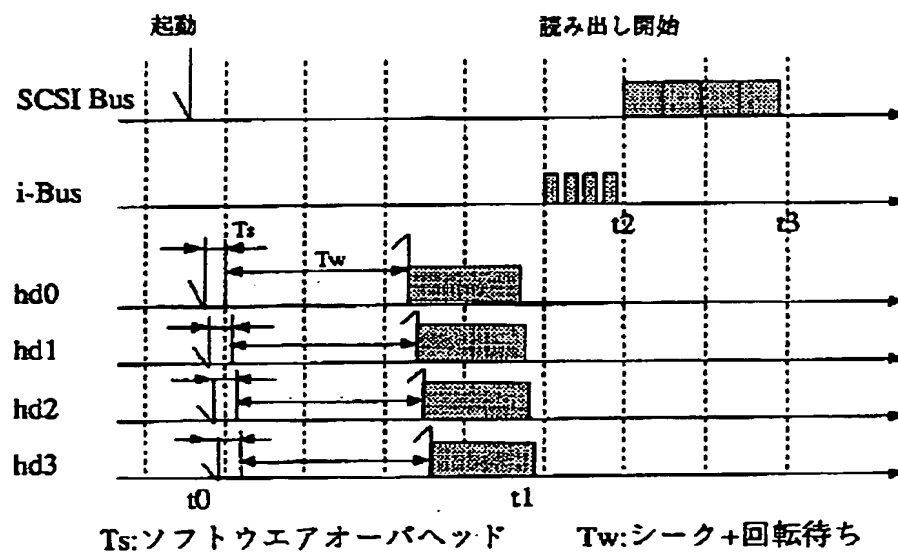
【図45】

図45



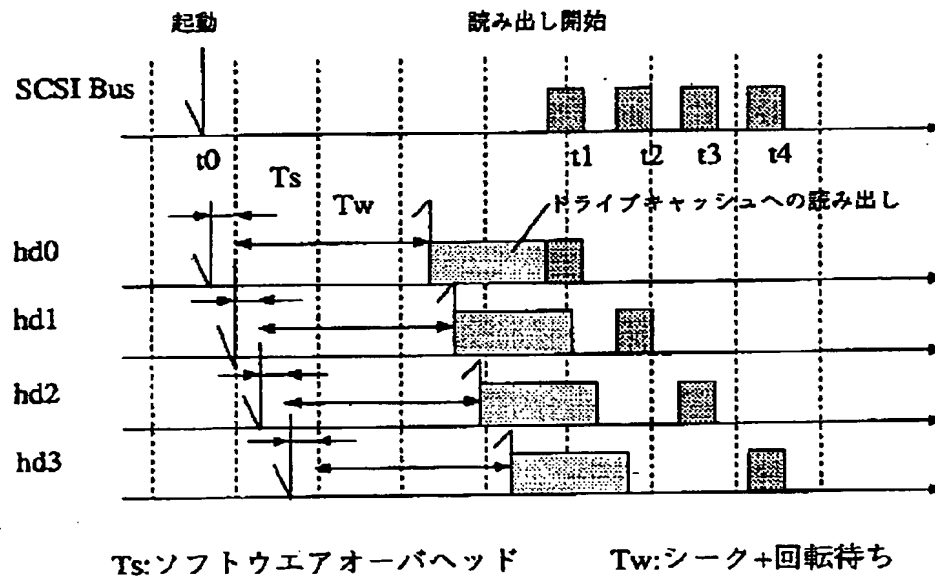
【図47】

図47



【図 4 9】

図 4 9



フロントページの続き

(72) 発明者 加藤 寛次
神奈川県川崎市幸区鹿島田890番地株式会
社日立製作所システム開発本部内

(72) 発明者 鈴木 広義
神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア事業部内
(72) 発明者 牧 敏行
神奈川県秦野市堀山下1番地 日立コンピ
ュータエンジニアリング株式会社内

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☒ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.